

A Physics-Guided Neural Network Dynamical Model for Droplet-Based Additive Manufacturing

Uduak Inyang-Udoh[✉], *Member, IEEE*, and Sandipan Mishra[✉], *Member, IEEE*

Abstract—This article develops a physics-guided data-driven model for the height evolution of parts printed in droplet-based additive manufacturing. The proposed model is a convolutional recurrent neural network (ConvRNN) whose structure is derived based on the physical understanding of mass conservation during the height evolution. Because of this physics-guided model structure, the model parameters obtained are invariant to the geometry of the printed part and thus portable from one geometry to another, the conditions on physical stability of the evolution translate directly to training stability of the neural network, and the data required to train this model are much less compared to a pure black-box model. These aspects of the model are validated experimentally on an inkjet 3-D printing setup. The proposed model outperforms a black-box off-the-shelf multilayer perceptron (neural network) by using about two orders of magnitude less data for training, at the same time delivering 1.7× smaller rms error on test data. The proposed model is also compared with a state-of-the-art reduced order linear model and shows 1.4× smaller rms error on test data. Finally, experimental results also underline that the model parameters learned are geometry invariant, that is, the model parameters trained on one geometry can be used to predict the height map evolution for other geometries without relearning.

Index Terms—Additive manufacturing (AM), neural network, stability.

I. INTRODUCTION

MANY additive manufacturing (AM) processes—such as inkjet 3-D printing, liquid metal jetting, and binderjet 3-D printing—involve the ejection of droplets (onto a substrate) that solidify to produce the desired part. These droplet-based AM processes enable the manufacture of intricate parts for applications in organic electronics, sensors and detectors, soft robots, and other biomedical articles [1]–[4]. In applications that require high precision and repeatability, it is critical to develop control-oriented models, which can be used for suitable feedforward input patterns and designing feedback control algorithms [5].

Manuscript received November 11, 2020; revised June 8, 2021; accepted October 14, 2021. Manuscript received in final form November 12, 2021. This work was supported in part by the NSF Data-Driven Cyberphysical Systems Award under Grant 1645648 and in part by the State of New York ESD/NYSTAR Program. Recommended by Associate Editor M. A. Grover. (Corresponding author: Uduak Inyang-Udoh.)

The authors are with the Mechanical, Aerospace and Nuclear Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: inyanu@rpi.edu; mishrs2@rpi.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCST.2021.3128422>.

Digital Object Identifier 10.1109/TCST.2021.3128422

Prior modeling studies on inkjet printing have focused on the physical dynamics of a single droplet or a few contacting droplets and have shown surface tension, inertia, fluid viscosity, and contact line motion to influence droplet deposition geometry and behavior [6]–[9]. However, these physics-based models are not suitable for applications that involve thousands of droplets deposited sequentially to create 3-D parts. This is because, given the complexity associated with modeling a single or few droplets, it is computationally very expensive to simulate dynamics at the part's geometry level from the first principles characterization of the fluid properties (including viscosity and surface tension). Furthermore, Thompson *et al.* [10] and Wu and Chiu [11] suggested that the bulk fluid behavior may vary with the number of droplets and time scale of deposition, and as a result, it may not be accurate to generalize such purely physics-based model for arbitrary printing tasks. Finally, while the models in these studies can be useful for predicting the printing outcome (geometry and bulk part properties), they are mostly unsuitable for feedforward and feedback control design.

In contrast with physics-based models, purely data-driven methods have also been used to determine the relationship between controllable process inputs and measurable output part quality for a variety of AM processes [12]. These methods are helpful for the selection of suitable process parameters, as well as in-process monitoring and control [4], [13], [14]. Moreover, they circumvent the characterization of fluid properties and the complexity of the physical dynamics and are relatively inexpensive computationally to implement. However, the data-driven methods have been mostly restricted to qualitative tasks, such as tuning printing parameters for satisfactory droplet size. Furthermore, these data-driven models (such as neural networks) lack interpretability (in terms of explaining the data representation inside the network) since no physical principles of the process are built into their formulation [15]. Consequently, large amounts of data are required to train a model capable of prediction under various scenarios and geometries.

Some prior work has addressed feedback control in droplet-based AM by using reduced order dynamic models [16]–[20]. The simplifications in these models enable closed-loop control at the geometry level of the part. Here, the entire geometry's height distribution is modeled as a superposition of the linear preceding distribution and current droplet(s) deposition. This

modeling approach incorporates just the basic physics of the AM process. Hence, they only require limited experimental data for the identification of (physically interpretable) parameters. However, these models have not been shown to perform well when the pattern of printing involves overlapping depositions and nonlinear effects, such as surface tension, are significant.

Inyang-Udoh and Mishra [24] formulated such a learning-based model structure, termed constrained-flow model, for inkjet 3-D printing and demonstrated that the recurrent neural network (RNN) model structure is amenable for feedback control. The spatial effect of droplet deposition was captured by a convolution and an activation function was proposed to determine the spatiotemporal evolution of the height profile. We build on this above work in the following ways: 1) post-deposition layer height shrinkage (due to curing, evaporation, and so on) is incorporated into the RNN model paradigm; 2) given the physical insight tied into the network’s design, analytical results regarding the open-loop stability of the model are derived; 3) the performance of the proposed model is experimentally validated and compared against a shallow

This article is organized as follows. In Section II, we describe the experimental setup and state the modeling problem and form of the model function to be developed. In Section III, previously developed reduced order models are previewed as context for the RNN model presented in Section IV. Section V presents the stability analysis for the model as well as discusses a gradient-based approach for learning the model. Results from training and validation of the model are presented in Section VI. Section VII presents the conclusions.

A. System Setup

Authorized licensed use limited to: The Libraries at Rensselaer Polytechnic Institute. Downloaded on May 03, 2022 at 14:56:03 UTC from IEEE Xplore. Restrictions apply.

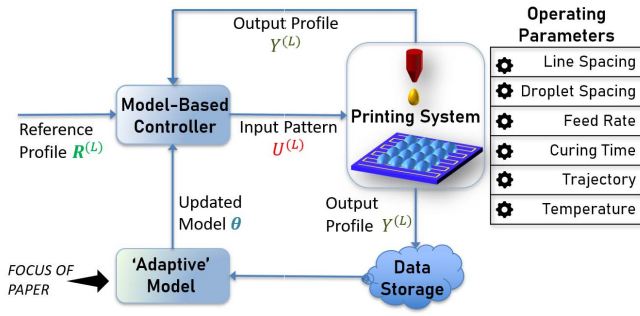


Fig. 2. 3-D inkjet printing scheme. For each layer, a reference profile based on the desired part geometry is fed to a controller, which generates suitable input sequence.

2-D laser sensor is used for sensing the height profile after each layer.

To print a part, first, the 3-D model of the part is sliced horizontally into layers and an associated motion path for the nozzle is generated along with droplet deposition locations. Motion stages move the build substrate, while the nozzle deposits droplets according to the input pattern. When all depositions for a layer are complete, the part is cured under UV light. Then, the laser sensor measures the height profile for the layer for feedback control. The process is repeated until all the layers are printed.

From a control standpoint, a layer, L , is resolved into an $\mathbb{R}^{n_x \times n_y}$ grid space to obtain the discretized reference height profile $R^{(L)} \in \mathbb{R}^{n_x \times n_y}$. The pattern of droplets $U^{(L)} \in \mathbb{R}^{m_x \times m_y}$ is the input to this system. The output $Y^{(L)} \in \mathbb{R}^{n_x \times n_y}$ (see Fig. 2) is measured after each layer is cured. A typical feedback control strategy is shown in Fig. 2. To implement such a control strategy, a model of the height profile evolution [i.e., a relationship between $U^{(L)}$ and $Y^{(L),(L+1)}$] is needed.

B. Problem Statement

The goal of this article is to learn the height evolution relationship from input–output data. Furthermore, this model must be formulated in such a way that the parameters are time and geometry agnostic and thus can be identified for any printed geometry, irrespective of printing area or droplet pattern. Given the model as $\hat{Y}^{(L+1)} = \Phi(\theta, Y^{(L)}, U^{(L)})$ where θ denotes all parameters of the model, $\hat{Y}^{(L+1)}$ is the model output of the next layer, and $Y^{(L)}$ is the current actual output, we shall determine a suitable structure for Φ and find θ^* that minimizes the error $\|Y^{(L+1)} - \hat{Y}^{(L+1)}(\theta)\|_2^2$ using the data obtained from printing. Furthermore, the parameters of θ^* to be learned should be explainable [15] and as frugal (small in number) as possible. Finally, given the spatiotemporal aspects of printing each layer, the evolution of the height profile should be modeled as a time sequence in the height profile space.

III. PRIOR WORK: MODELING APPROACHES FOR DROPLET-BASED AM

As highlighted in Section I, several reduced-order models for droplet-based AM processes have been developed to enable

feedback control. We briefly discuss these model paradigms as they springboard the model development to follow.

A. Droplet Superposition

Several geometry-level models have expressed the print topology evolution as a superposition of the individual droplets height distributions [17], [20]. Assuming that the distribution is constant, this may be written as $\hat{Y}^{(L+1)} = Y^{(L)} + b * U^{(L)}$, which is a 2-D convolution of $b \in \mathbb{R}^{p \times p}$, the impulse response due to a unit droplet input, over the input space $U^{(L)}$. This impulse response may be assumed Gaussian [17], spherical [16], [20], or the average of actual sample droplets [25] or may be identified from a high-fidelity simulation of the printing process [19].

B. Droplet Interaction and Flow

Other models incorporate the effect of the liquid material flow. With this outlook, flow between neighboring grid points is characterized as a function of relative heights between the points [20], [26]. Let $H_t \in \mathbb{R}^{n_x \times n_y}$ be the height profile at time t in the printing process and $h_t \in \mathbb{R}^n$, $n = n_x \times n_y$, be the corresponding vector, and then, the change in height at point (or node) i due to flow from neighboring nodes is given by [15]

$$\Delta h_t(i) = - \sum_{j \in \mathcal{N}_i} K_{ij}(h_t(i) - h_t(j)) + B_t(i)u_t \quad (1)$$

where $-\sum_{j \in \mathcal{N}_i} K_{ij}(h_t(i) - h_t(j))$ represents the height change due to flow from all other nodes j in the neighborhood $\mathcal{N}_i$ of node i , $K_{ij} > 0$ is the flowability parameter that describes how much the liquid will flow based on the neighbors' height differences, and u_t is the droplet volume at time t . $B_t(i)$ is the corresponding height increase at node i caused by a unit size droplet, that is, $B_t \in \mathbb{R}^n$ is a vector containing a unit droplet's shape and location. In state-space notation, the evolution is written as

$$h_{t+1} = A_t h_t + B_t u_t. \quad (2)$$

$A_t = (I - DK_t D^T)$ is the state matrix that captures the effect of liquid flow from higher to lower grid points, $D \in \mathbb{R}^{n \times n_t}$ being the incidence matrix for the grid with n_t links (see Fig. 4), and $K \in \mathbb{R}^{n_t \times n_t}$, a diagonal positive (semi)definite matrix containing flowability parameters determined empirically. K denotes how much flow would occur in one time step due to a unit height difference. K_t is used since the matrix must be continuously updated from time step to time step to dissociate links active with flow from those with no flow. As will be discussed in Section IV, the flowability parameter and incidence matrix are utilized in the model presented in this article.

IV. RNN MODELING WITH PHYSICS-GUIDED STRUCTURE

In this section, a convolutional recurrent neural network (ConvRNN) model that is based on physical knowledge of the printing process is developed. We begin by discussing the basics of an RNN and show how physical understanding of the ink-jetting process can be used to construct the network.

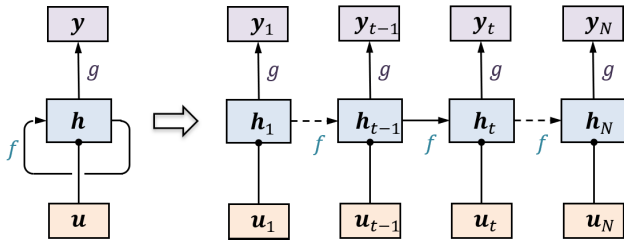


Fig. 3. RNN as parameterized in (3).

The result is a so-called constrained-flow model for the printing process that is time-invariant, whose parameters are geometry independent and thus can be used for different layer patterns.

A. Recurrent Neural Network

RNNs are neural network models used for modeling time-series dynamical models [27], [28] for a variety of applications. A general formulation of the network can be written as

$$\begin{aligned} h_t &= f(h_{t-1}, u_{t-1}, \theta_1) \\ y_t &= g(h_t, \theta_2) \end{aligned} \quad (3)$$

where u_{t-1} and h_{t-1} are input and state of the system at time step $t-1$, respectively, y_t is the output at time step t , and $\{\theta_1, \theta_2\} = \theta$ is the set of the model parameters (see Fig. 3). Note that the input and state are time-dependent, while the parameters are time-invariant.

RNNs are typically parameterized as

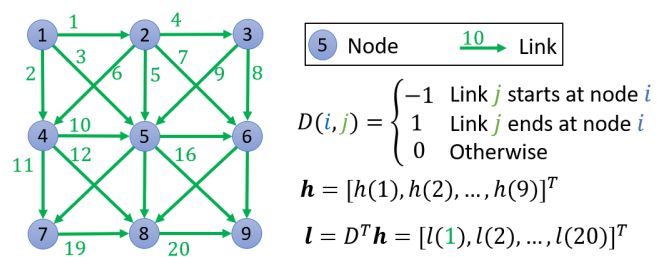
$$\begin{aligned} h_t &= W_h \phi_1(h_{t-1}) + W_u u_{t-1} + w_0 \\ \hat{y}_t &= \phi_2(W_y \phi_1(h_t) + v_0) \end{aligned} \quad (4)$$

where W_h is the recurrent or state weight matrix, W_u is the input matrix, w_0 is the state bias, and W_y and v_0 are the output weight and bias, all elements of the parameter set θ , respectively. ϕ_1 and ϕ_2 are elementwise nonlinear function typically termed activation functions. θ is unknown and thus must be learned or identified from input–output data pairs $\{u_t, y_t\}_{t=1-N}$. A cost function (loss function) measures the performance of the network on some given regression or classification task. The parameters are learned to minimize this cost.

B. Mapping Inkjet Process Dynamics to ConvRNNs

The printing process involves jetting droplets one after the other; the part's height evolves as the droplets are deposited. Hence, the process can be described as a time series and an RNN is suitable for modeling it. The generic RNN formulation can be tailored, guided by the physics of the printing process, as follows.

- 1) The droplet position and volume at instant t constitute the input to the time series. Moreover, instead of multiplication by an input matrix as in (3), we convolve a weight matrix about the input space based on the interpretation presented [17]. This isolates the spatial

Fig. 4. Incidence matrix D transforms the height profile at grid points (or nodes) into height differences (links). Each node is linked to its immediate neighbors.

property of the input from the input weight parameter and reduces its dimension.

- 2) The nonlinear activation function ϕ_1 is chosen to incorporate intuition about the basic dynamics that influence height evolution. First, the height profile at each time step is extrapolated to a higher dimensional space. In this space, the heights at grid points are transformed into height differences between neighboring grid points using an incidence matrix (see Fig. 4). Furthermore, because of surface tension, a minimum threshold of height difference is needed to initiate flow. Hence, an element-wise function is applied to embody this flow threshold. Finally, the output of the function is returned projected to the original space.
- 3) In many droplet-based AM systems, it is difficult to make measurements during the in-layer deposition. The profiles are only measured after deposition. This means that the output data for the network are only available after the last time step in a layer. Hence, the output in (3) is only computed at $t = L \times N_L$, where L is the layer number. Furthermore, another activation function ϕ_2 is used to model the reduction in part volume due to curing or evaporation.

The result of this formulation is a ConvRNN [29] where the following conditions hold.

- 1) The spatial dynamics of the process is captured by a 2-D convolution over the input.
- 2) The temporal dynamics is captured by the nonlinear state evolution.
- 3) The effect of postprocessing after each layer (curing) on the height profile is captured by a nonlinear output function.

C. ConvRNN Model

Following the reasoning above, the height evolution for layer L can be written as:

$$\begin{aligned} h_t &= \phi_1(h_{t-1}) + W_u u_{t-1}, \quad t \in [1, N_L] \\ \hat{y}^{(L)} &= \phi_2(\phi_1(h_{N_L}) + v_0) \\ h_0 &= \hat{y}^{(L-1)} \end{aligned} \quad (5)$$

where $h_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$, $m = m_x \times m_y$, are the network's internal state (or height) and input. The function $\phi_1(h_t)$ reflects the height change at time step t

$$\phi_1(h_t) \triangleq h_t - D\sigma(KD^T h_t). \quad (6)$$

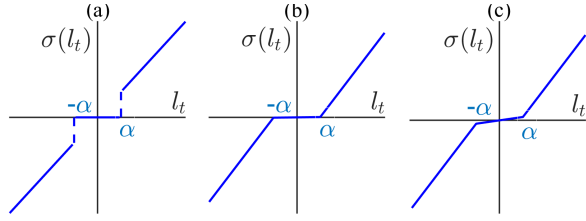


Fig. 5. (a) Hard thresholding. (b) Soft threshold function. (c) Leaky soft threshold function. Potential activation functions to capture surface tension inhibition to flow across links. All three functions ensure mass conservation since they are odd, $\sigma(-l_t) = -\sigma(l_t)$.

Note that $D \in \mathbb{R}^{n \times n_l}$ transforms the height profile vector into height differences across links (Fig. 4). These differences are then weighted by K . In general, K is a diagonal matrix whose entries correspond to the flowability associated with each link, $K_{i,j}$, as described in (1). Therefore, together, $KD^T h_t$ is the effective flow across links at time t due to height differences across each link, $D^T h_t$.

σ thresholds the minimum effective height difference needed across a link to cause flow (capturing some of the surface tension effects). Fig. 5 shows the potential thresholding functions. The soft threshold of Fig. 5(b) is preferred to the function of Fig. 5(a) as it is continuous and holds a threshold value explicitly in its formulation. Yet, as is highlighted in Section V, identifying the model parameters by gradient-based methods requires the stability of the function's gradient [30], [31]. The function of Fig. 5(c), which we term leaky soft threshold, allows for gradient stability over long time steps and hence is used in this work. Denoting $KD^T h_t$ by l_t , the activation function can be written as

$$\sigma(l_t(i)) \triangleq \begin{cases} l_t(i) - (1 - \delta)\alpha, & \text{if } l_t(i) > \alpha \\ \delta l_t(i), & \text{if } -\alpha \leq l_t(i) \leq \alpha \\ l_t(i) + (1 - \delta)\alpha, & \text{if } l_t(i) < -\alpha \end{cases} \quad (7)$$

where α is the threshold below which height differences cause no flow; $\delta > 0$ ensures that the gradient of the activation function is not zero. Unlike with the graph-based model mentioned in [20], K no longer needs to be updated with time. Since parts are cured only at the end of each layer, we assume that the flow is isotropic and write $K = \kappa I$. Note that $\phi_1(h_t)$ ensures conservation of mass, that is, $\mathbf{1}^T \phi_1(h_t) = \mathbf{1}^T h_t$.

$W_u u_{t-1}$ corresponds to the height profile increment caused by input droplet deposition at time step t . $U_t \in \mathbb{R}^{m_x \times m_y}$, $t \in [0, N_L - 1]$ is the admissible inputs at time step t . u_t is the vectorized form of this input. Note that U_t is sparse, holding only a nonzero entry at (x, y) positions ($x \in \{1, \dots, n_x\}$, $y \in \{1, \dots, n_y\}$) where depositions occur at t , that is, $\sum_{t=0}^{N_L-1} U_t = U^{(L)}$. W_u is the Toeplitz matrix corresponding to kernel $b \in \mathbb{R}^{p \times p}$ in the 2-D convolution, $b * U_t$, that is, $W_u u_t = \text{vec}(b * U_t)$. Hence, we may equivalently implement a sparse 2-D convolution and vectorize the resulting matrix. Again, unlike the time-dependent input matrix in (2), W_u is time-invariant. It is well known that the volume of the part shrinks after curing [15]. We make provision for this shrinkage in the model in the output step. A negative scalar bias v_0 is added elementwise to the height profile output after the last

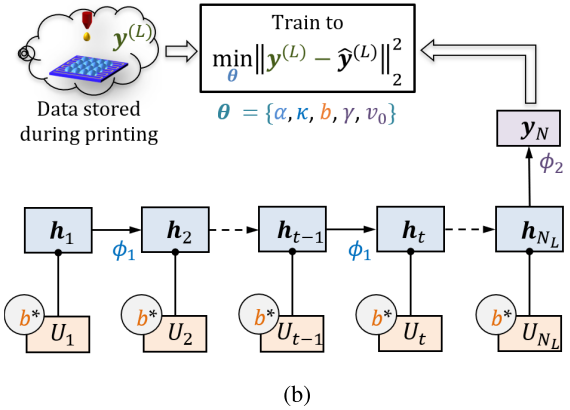
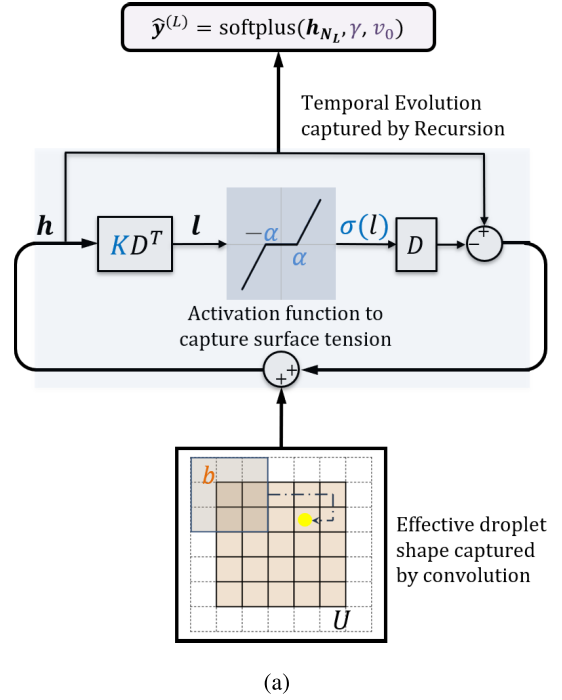


Fig. 6. Illustration of time step to time step height evolution model as an RNN. (a) Height evolution as a recursion. At each time step, the kernel w_u is convolved over the current input space, which holds entry only at the current deposition spot (yellow spot in the figure), and the result is added to the evolving height distribution. (b) Height evolution unrolled as an RNN.

time step, and then, the following generic softplus function is applied as shown:

$$\phi_2(h^{(L)}(i)) \triangleq \log(\gamma + \exp h^{(L)}(i)). \quad (8)$$

The softplus function is parameterized by γ , which is nonnegative. Note that this function also ensures that the output profile is nonnegative. The ConvRNN-based inkjet printing model is summarized in Fig. 6.

Note: First, while the ConvRNN model follows the ink-jetting paradigm of droplets deposition in sequence, the time step of the model need not correspond to that of actual deposition. For example, if the input space exceeds the output space, then we may sample (or coarsen) the input grid space such that the total number of model time steps is only a fraction of actual printing time steps. As such, the input

convolution kernel b , here, is to only embody the droplets' behavior in the context of the final resulting height profile, rather than the shape of a single deposited droplet (or a droplet impulse response).

Second, in addition to time invariance, this model is independent of the geometry or printing trajectory being followed: as such, the model is left only to learn the fluid behavior that affects the height evolution. This makes the model versatile and allows for little training data. Furthermore, the model may be employed for cases with either single or multiple ink-jetting nozzles.

V. MODEL ANALYSIS

Since the height evolution model in (6) is structured based on conservation of mass, we expect it to be input–output stable in continuous time. In this section, a formal discrete-time stability analysis for the height evolution ConvRNN model is presented. In addition, analytical expressions of gradients for training the network through backpropagation are presented. As shown in [24], providing analytical gradient expressions to the solver expedites online learning and control. Moreover, with these expressions, we can draw connections between the physical structure of the height evolution and the training of the network. Indeed, we show that the same sufficiency condition for the stability in the height evolution holds for the stability of the ConvRNN training.

A. Stability Analysis

Assuming zero input in (5), the height evolution becomes: $h_t = \phi_1(h_{t-1})$. Note that the function ϕ_1 can be written as

$$\phi_1(h_t) = (I - DK_\delta(h_t)D^T)h_t \quad (9)$$

where $K_\delta(h_t) = \text{diag}(k_\delta)$ and each diagonal element is defined as

$$k_\delta(i) \triangleq \begin{cases} \kappa(1 - (1 - \delta)\alpha/l_t(i)), & \text{if } l_t(i) > \alpha \\ \delta\kappa, & \text{if } -\alpha \leq l_t(i) \leq \alpha \\ \kappa(1 + (1 - \delta)\alpha/l_t(i)), & \text{if } l_t(i) < -\alpha. \end{cases} \quad (10)$$

Stability Lemma: First, let $\rho(DD^T)$ be the spectral radius of the Laplacian DD^T , and the height evolution dynamics in (5) is stable (i.s.L) about the equilibrium origin $h_0 = \text{constant}$ if

$$0 < \kappa < 2/\rho(DD^T). \quad (11)$$

Second, furthermore, for a (gridded) graph of the form shown in Fig. 4

$$0 < \kappa < 1/6. \quad (12)$$

Proof: First, consider the autonomous evolution, $h_{t+1} = \phi_1(h_t)$. Define the Lyapunov function $V(h_t) = h_t^T h_t$

$$\begin{aligned} V(h_{t+1}) - V(h_t) &= h_t^T ((I - DK_\delta(h_t)D^T)^T (I - DK_\delta(h_t)D^T) - I) h_t \\ &= -h_t^T ((2I - DK_\delta(h_t)D^T) DK_\delta(h_t)D^T) h_t. \end{aligned} \quad (13)$$

We now show that $(2I - DK_\delta(h_t)D^T) DK_\delta(h_t)D^T \geq 0$. Consider $DK_\delta(h_t)D^T$: from (10) and (11), $k_\delta(i) \geq \delta\kappa > 0 \forall i$; hence,

$$DK_\delta(h_t)D^T \geq 0. \quad (14)$$

Next, let $Q = 2I - \kappa DD^T$. Each eigenvalue $\lambda(Q) = 2 - \kappa\lambda(DD^T)$. From (11), $\kappa < 2/\rho(DD^T)$, hence, all $\lambda(Q) > 0$. By definition (10), $k_\delta(i) < \kappa \forall i$; thus,

$$2I - DK_\delta(h_t)D^T \succ Q \succ 0. \quad (15)$$

Since $2I - DK_\delta(h_t)D^T$ and $DK_\delta(h_t)D^T$ are symmetric and commute, $(2I - DK_\delta(h_t)D^T) DK_\delta(h_t)D^T \geq 0$; therefore, $V(h_{t+1}) - V(h_t) \leq 0, \forall h_t \neq 0$ as required for stability.

Second, it is shown in [33] that the Laplacian is conservatively bounded as follows:

$$\begin{aligned} \rho(DD^T) &\leq \max\{d_i + d_j - |\mathcal{N}_i \cap \mathcal{N}_j| \\ &\quad : 1 \leq i < j \leq n, (h(i), h(j)) \in E\} \end{aligned} \quad (16)$$

where d_i is the degree of $h(i)$, $\mathcal{N}_i$ is the set of neighbors of $h(i)$, and E is the edge (or node) set (which has n_l elements) of the graph. For the graph (as connected in Fig. 4) with any number of nodes n , the bound implies $\rho(DD^T) \leq 12$, and hence, given (11)

$$0 < \kappa < 1/6.$$

B. Analytical Gradients for Training the ConvRNN Model

We now derive analytical expressions of the gradients needed for training the ConvRNN and demonstrate that the criteria for open-loop stability in (11) is indeed the same for guaranteeing stability in training the network. Fig. 6 shows the model recursion and how it may be unrolled as an RNN with output only at the network's final time step. Each measured layer provides additional data for training the RNN. For learning the optimal parameter set $\theta = \{\alpha, b, \kappa, \gamma, v_0\}$, the 2-norm of model error at the L th layer is minimized, i.e.,

$$\begin{aligned} \theta^* &= \underset{\theta}{\operatorname{argmin}} \|y^{(L)} - \hat{y}^{(L)}(\theta)\|_2^2 \\ \text{s.t. } \theta_{\min} &\leq \theta \leq \theta_{\max} \end{aligned} \quad (17)$$

where $y^{(L)}$ is the vectorized measured height profile of the L th layer and $\hat{y}^{(L)}(\theta)$ is the corresponding computed height of the RNN's forward pass; $\theta_{\min}$ and $\theta_{\max}$ are, respectively, the maximum and minimum values the parameters of θ may assume. Let the 2-norm of model error $\|y^{(L)} - \hat{y}^{(L)}(\theta)\|_2^2$ be denoted by $E^{(L)}$, and the gradient of $E^{(L)}$ with respect to the variable $\theta \in \{\alpha, \kappa\}$ over P preceding layers is

$$\frac{\partial E^{(L)}}{\partial \theta} = \sum_{L-P \leq i \leq L} \sum_{1 \leq t \leq N_L} \frac{\partial E^{(L)}}{\partial \hat{y}^{(L)}} \frac{\partial \hat{y}^{(L)}}{\partial h_t} \frac{\partial h_t}{\partial \theta}. \quad (18)$$

Here,

$$\begin{aligned} \frac{\partial E}{\partial \hat{y}^{(L)}} &= 2(y^{(L)} - \hat{y}^{(L)})^T \\ \frac{\partial \hat{y}^{(L)}}{\partial h_t} &= \frac{\partial \hat{y}^{(L)}}{\partial h_{N_L}} \frac{\partial h_{N_L}}{\partial \hat{y}^{(L-1)}}, \dots, \frac{\partial \hat{y}^{(i)}}{\partial h_{N_i}} \frac{\partial h_{N_i}}{\partial h_t} \end{aligned}$$

where

$$\begin{aligned} \frac{\partial \hat{y}^{(i)}}{\partial h_{N_i}} &= \text{diag}(1/(1 + \gamma \exp \mathbf{1} v_0 - h_{N_i})) \\ \frac{\partial h_{N_i}}{\partial h_t} &= \prod_{N_i-1 > j \geq t} (I - D \text{diag}(\sigma'(l_j)) K D^T) \end{aligned}$$

and

$$\begin{aligned}\frac{\partial h_t}{\partial K} &= -D \text{diag}(\sigma'(l_{t-1})) D^T h_{t-1} \\ \frac{\partial h_t}{\partial \alpha} &= -D \text{diag}(\sigma'(l_{t-1})) \sigma'(\alpha) \\ \sigma'(l_t) &= \begin{cases} 1, & \text{if } l_t(i) > \alpha \\ \delta, & \text{if } -\alpha \leq l_t(i) \leq \alpha \\ 1, & \text{if } l_t(i) < -\alpha \end{cases} \\ \sigma'(\alpha) &= \begin{cases} \delta - 1, & \text{if } l_t(i) > \alpha \\ 0, & \text{if } -\alpha \leq l_t(i) \leq \alpha \\ 1 - \delta, & \text{if } l_t(i) < -\alpha. \end{cases}\end{aligned}$$

Similarly, the gradient of $E^{(L)}$ with respect to $\theta \in \{\gamma, v_0\}$ is

$$\frac{\partial E^{(L)}}{\partial \theta} = \frac{\partial E^{(L)}}{\partial \hat{y}^{(L)}} \sum_{L-P \leq i \leq L} \frac{\partial \hat{y}^{(L)}}{\partial \hat{y}^{(i)}} \frac{\partial \hat{y}^{(i)}}{\partial \theta} \quad (19)$$

where

$$\begin{aligned}\frac{\partial \hat{y}^{(L)}}{\partial \hat{y}^{(i)}} &= \frac{\partial \hat{y}^{(L)}}{\partial h_{N_L}} \frac{\partial h_{N_L}}{\partial \hat{y}^{(L-1)}}, \dots, \frac{\partial h_{N_{i+1}}}{\partial \hat{y}^{(i)}} \\ \frac{\partial \hat{y}^{(L)}}{\partial \gamma} &= 1/(\gamma + \exp h_{N_L} - \mathbf{1}v_0) \\ \frac{\partial \hat{y}^{(L)}}{\partial v_0} &= -1/(1 + \gamma \exp \mathbf{1}v_0 - h_{N_L}).\end{aligned}$$

The gradient of the error with respect to the kernel b is

$$\frac{\partial E^{(L)}}{\partial b} = \sum_{L-P \leq i \leq L} \sum_{1 \leq j \leq N_L} \text{vec}^{-1} \left(\frac{\partial E^{(L)}}{\partial \hat{y}^{(L)}} \frac{\partial \hat{y}^{(L)}}{\partial h_t} \right) * \text{rot}_{180}(U_i) \quad (20)$$

where vec^{-1} denotes the vector matricization and rot_{180} denotes a 180° rotation. Given the gradient directions, the constrained optimization in (17) may be implemented with a barrier function method as the interior point method [34]. Bounds on θ are necessary for the parameters α, κ , and γ : α and γ must be nonnegative and κ must satisfy (11).

Relationship Between Training and Physical Stability: One major consideration in training an RNN is the possibility of exploding gradients [35]. In particular, the term $(\partial h_{N_L} / \partial h_t)$ may explode as N_L becomes large. Since $I - D \text{diag}(\sigma'(l_j)) K D^T$ is symmetric, $(\partial h_{N_L} / \partial h_t)$ is kept from exploding (or stable) if spectral radius $\rho(I - D \text{diag}(\sigma'(l_j)) K D^T) \leq 1$. Interestingly, this condition for stability automatically holds given (11), $0 < \kappa < 2/\rho(DD^T)$, that is, the stability of the height evolution dynamics. This implies that training stability of the model structure translates to physical stability of the evolution dynamics.

Remark 1: Note also that, the network never “dies” [30], [36], i.e., it never becomes a constant function as $\sigma'(l_t) > 0 \forall t$.

VI. EXPERIMENTAL RESULTS

This section presents the experimental validation of the proposed physics-guided ConvRNN model. First, the performance of the proposed ConvRNN model trained on a small dataset is

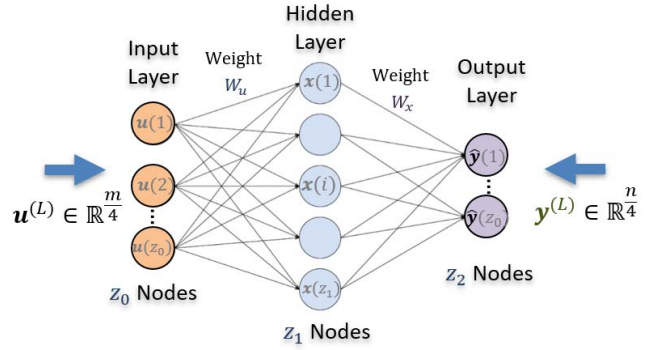


Fig. 7. Structure of the Z_0 - Z_1 - Z_2 MLP NN used for regression.

TABLE I
DETAILS OF PRINTED GEOMETRIES AND MLP NETWORK

Item	Detail
Data Points (Layers)	$\{u^{(L)}, y^{(L)}\}$ pairs
Printed geometries	I, U, r, L, n, T, -, C, +, $\square$
Training data size	216 data points (of all but two of the above geometries); augmented to 864 by rotation
Test data size	36 data points (of T and U geometries)
Weights	$W_u \in \mathbb{R}^{8 \times 256}$; $W_x \in \mathbb{R}^{256 \times 8}$
Layers	256-8-256
Activation function	arctan sigmoid
Loss function	Mean Squared Error

compared to that of a shallow neural network trained using a much larger dataset. Comparison is also made with the linear model discussed in Section III. Next, the ConvRNN is trained and validated under different printing scenarios, to ascertain that despite the printing method, the features captured by the model are geometrically independent. Significant improvement in accuracy on both the training and test data in all printing cases is demonstrated. Furthermore, the physics-guided model requires far less training data compared to a shallow (black-box) neural network model.

A. Performance Comparison of ConvRNN Model With Multilayer Perceptron Neural Network and Linear Dynamical Model

We evaluate the performance of the proposed ConvRNN model in comparison to an off-the-shelf shallow multilayer perceptron (MLP) network [37] (see Fig. 7) as well as the linear model of (2) for context. We printed 84 parts with five or six layers of various dimensions of simple geometric shapes. Each part was printed as a frustum such that though all the layers have the same cross-sectional geometry, each layer is dimensionally smaller than the preceding one. Layer heights ranged between 30 and 45 μm , printed with droplets in the volume range of 0.4–0.6 nL. Geometries and details of the training/test data used are given in Table I. We set aside 12 of the parts (having either a “T” or “U” shape) for testing and the rest for training. Three layers of height measurements of each part were selected as output data points, and the corresponding input pattern was used as the network’s input: 36 layers in total for testing and 216 layers for training. These are used as

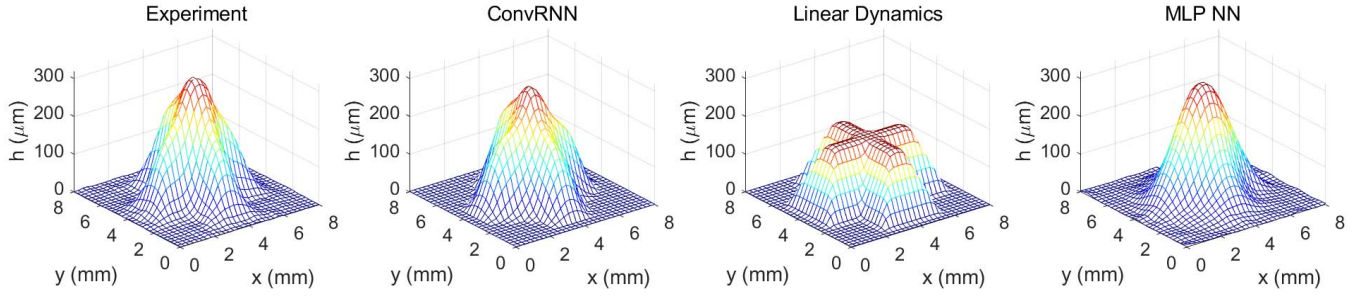


Fig. 8. Comparison of sections of a sample part from the training dataset of the proposed ConvRNN model, MLP NN, and linear dynamical model.¹

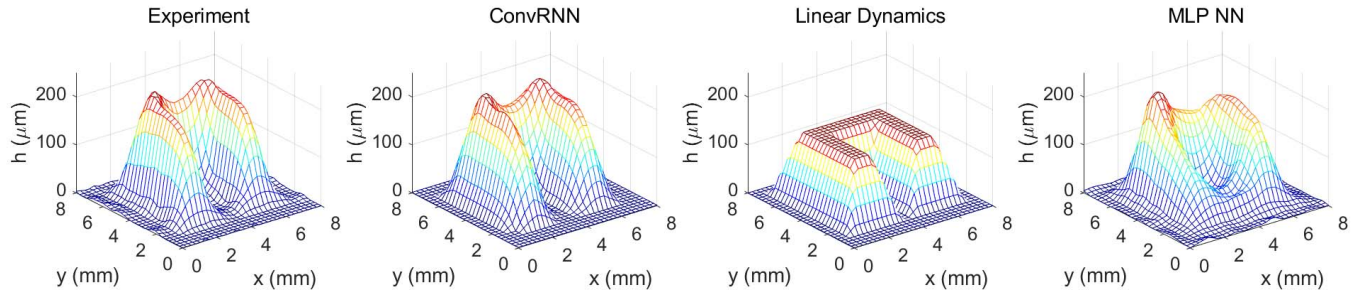


Fig. 9. Comparison of sections from the predictions of the trained proposed ConvRNN model, linear model, and a trained MLP NN on a sample test part.

follows for the MLP, linear model, and proposed ConvRNN model.

1) *MLP Model*: Each of the 216 training data points (each data point corresponds to a dataset from an entire layer) is rotated in three orthogonal directions to augment the dataset to 864 for training the MLP. The augmentation is to emphasize the spatial correlation between the input and output data pairs. Each data pair is on a 64×64 grid. We downsample each pair by 2-D linear interpolation so that the length of the vector is 256. This keeps the dimensionality of the data lower than the data size. In addition to saving computational cost, this helps us learn only the overall liquid material behavior and remove details that may only be unique to a particular printing session. The training was carried out by backpropagation on MATLAB with the hyperparameters summarized in Table I. The network performance is evaluated using the 36 (downsampled) test data points (layers).

2) *Linear Dynamic Model*: All 216 data points (data from 216 layers) are used for system identification of the linear model in (2). The reader is referred to [20] for details. The identified model is then used for prediction on the test datasets.

3) *ConvRNN Model*: The proposed ConvRNN model is trained on only three data points (three layers) from the same training dataset (see Table II for detail) but

TABLE II
DETAILS FOR CONVRNN

Item	Detail
Data Points (Layers)	$\{u^{(L)}, y^{(L)}\}$ pairs
Training data size	3 data points (of the + and n geometries above)
Test data size	36 data points (as in Table I above)
Parameters	$\alpha, b, K, \gamma, v_0$

TABLE III
EVALUATION OF PERFORMANCE (BASED ON RMS ERROR) ON DATASET WITH MLP NEURAL NETWORK, LINEAR MODEL OF (2), AND THE PROPOSED CONVRNN MODEL

Method	Data Size (Layers)		RMS Error (μm)	
	Training	Test	Training	Test
MLP NN	216 ($\times 4$)	36	19.4	34.2
Linear Model	216	36	31.6	27.9
ConvRNN (64×64 grid)	3 3	36 36	10.4 13.4	18.2 19.2

evaluated for performance using all 36 downsampled test data points.

Results from the training are summarized in Table III. Observe that on the training dataset, the MLP performs much better than the linear model. Fig. 8 shows one of the training data points a five-layered cross-shaped part. Note that this data point is common to all three models, that is, it is one of the three data points used to train the convRNN. The middle x - and y - cross sections of this sample part are shown in Fig. 10(a). These figures demonstrate that while the linear model yields a uniform surface profile, the MLP

¹Mesh grid used here is for graphical esthetics and does not represent the actual grid resolution.

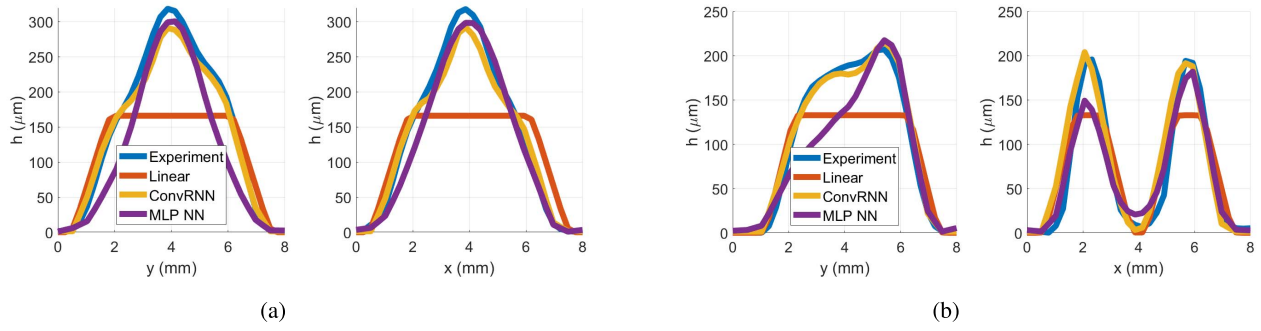


Fig. 10. Cross sections of cross-shaped part used for training and T-shaped part for testing. (a) Middle cross sections in both x - and y -directions. (b) Middle cross section in the x -direction and cross section through the left arm through the y -direction.

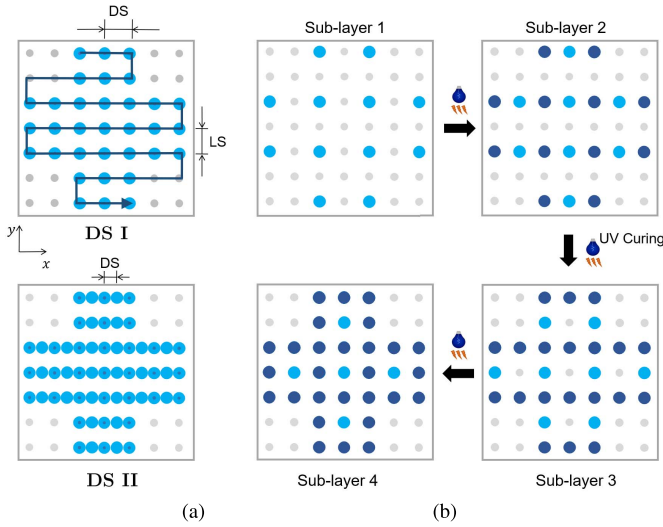


Fig. 11. Illustration of strategies used for printing a cross-shaped section on an $n_x \times n_y = 7 \times 7$ grid. The blue dots represent deposition; darker blue colors imply the deposition has been cured. (a) Top: printing trajectory follows a raster path; the distance between lines of the raster, that is, LS is equal to the grid resolution and the interval at which droplets are deposited (DS, Spacing A). Bottom: droplets are deposited at intervals (DS) one-half the grid resolution. (b) Each layer is built in four steps following the same raster trajectory (Spacing B). The section is cured after each step. LS and DS are twice grid resolution at individual sublayers.

captures the outward curvature. However, with the MLP, the geometrical structure is lost (the distinct cross arms are lost). The ConvRNN model, on the other hand, is able to capture the outer curvature while also retaining geometric fidelity.

Furthermore, because the learned MLP network weights (while capturing nonlinearities) are not geometrically independent, the improved rms error² on the training data does not translate to the test data; in fact, the performance is poorer than that of the linear model (Table III). This indicates that although the multilayer perceptron neural network (MLP NN) learns to accurately map the input to the output on the training part, the mapping is overfit. The overfitting problem is not simply resolved by using a relatively small number of nodes in the hidden layer. Notice that the hidden layer of the MLP contains only eight nodes in contrast to the 256 in the input

²The rms error is calculated over the entire grid space.

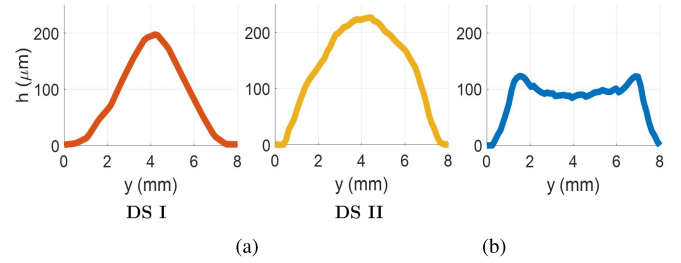


Fig. 12. Comparison of output profile shapes for printing Spacings A and B. The liquid material assumes different profiles. (a) Bell-shape bulge and parabolic shape under DS I and II, respectively (Spacing A). (b) Concave shape for Spacing B.

and output layers. Decreasing the number of hidden nodes would only render the network ineffective for the complexity of the regression problem [38]. Rather, improved performance of the NN calls for increased nodes and, consequently, needs a much larger amount of data for training.

On the other hand, from Table III, it can be noted that even with such small training dataset, the ConvRNN model performs as well on the test dataset as the shallow NN does on its training dataset. A sample from the test dataset is shown in Fig. 9 with cross sections in Fig. 10(b). The sample is a four-layer U-shape part. This figure highlights how qualitatively well the proposed model predicts the actual print. Furthermore, it is key to note that the NN is unable to predict the part's distinctive features. The linear model traces the shape distinctively, but it is unable to capture the characteristic bulging artifact of the print.

Remark 2: In this section, the 64×64 data pairs were treated as images and downsampled to 16×16 to keep the data dimensionality lower than the data size for the MLP NN. For comparison, the downsampled input and output pairs were assumed true input and output for the linear model and the convRNN. For the 16×16 data pairs, a 5×5 b matrix is used for the convRNN. Though downsampling is important for the MLP NN, in downsampling the original 64×64 grid space, the input resolution is lost. Hence, the convRNN training was repeated for the original 64×64 with a larger b matrix (13×13) and yielded similar test errors (shown in Table III). In Section VI-B, we use the original grid resolution for training.

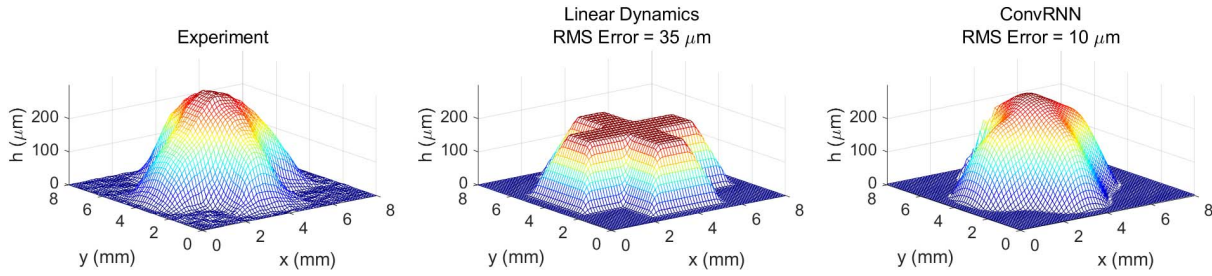


Fig. 13. Comparison between the linear dynamic model and the proposed model for cross-shaped part used for training with reduced DS [DS II in Fig. 11(a)].

B. Performance Under Different Printing Scenarios

Fig. 11(a)-I shows the manner in which the parts in Section VI-A were printed. The printing is carried out in a raster manner and the spacing between droplets is equal to the spacing between the raster lines. We explore other printing spacing scenarios that may alter a height profile. One is decreasing the droplet spacing (DS), as shown in Fig. 11(a)-II. Here, the DS is half the line spacing (LS), but each droplet has the same volume. Obviously, this increases the total volume of each layer. However, it also alters the shape of the height profile for a cross-shaped part printed in this manner. The additional volume allows for a smoother height distribution (less surface unevenness as there tends to be better droplet coalescence all across the layer). The resulting difference in height profile is highlighted in Fig. 12. The figure shows the cross-sectional profiles of three-layer cross-shape parts printed, as shown in Fig. 11. It can be observed that the bell-shaped profile of the part with DS I becomes parabolic with DS II.

Fig. 11(b) shows another printing scenario that results in a flatter height profile. The input resolution here is the same as that of Fig. 11(a)-I, but now, curing is carried out at instances within a layer. The LS and DS before each curing instance (or sublayer) is twice that of Fig. 11(a)-I. This strategy, though more time-consuming, tends to produce a more distinct and accurate height profile. The droplets are better spaced out and nearly isolated before the first curing step. However, as more droplets fill in the grid spots, surface-tension-driven droplet interaction occurs and the overall section profile becomes concave [Fig. 12(b)]. We evaluate the performance on these printing scenarios that result in markedly different height profiles to show that the geometry independence is, in fact, separately kept for each of these printing conditions.

Remark 3: The grid resolution is determined by the fineness of the output profile. Meanwhile, the LS and DS determine the input resolution. In practice, the output resolution is limited by the height profile sensor, while the LS and DS are limited by the resolution of the motion stages.

1) *Change in DS:* We verify that we can capture the height profile behavior while printing with the reduced DS as shown in Fig. 11(a)-II with a significant level of geometric independence. We train on only one data point, the four-layered cross-shaped part shown in Fig. 13. The part is printed with an LS of 125 μm and DS of 62.5 μm . Given that the droplets are deposited in between grid spots (of the grid

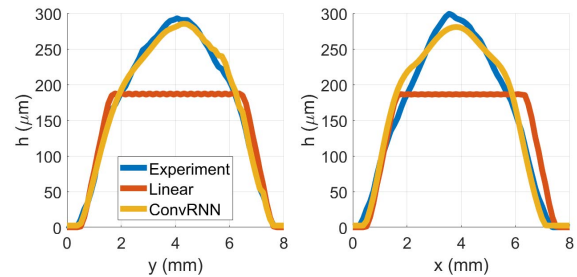


Fig. 14. Section of cross-shaped part used for training.

resolution), the kernel learned captures the effect of those two adjacent droplets at each grid point. On self-validation, the identified parameters yield an rms error of 10 μm on the cross-shaped frustum, an improvement in accuracy of 71.4% over the prediction of the linear dynamic model (Fig. 13). Furthermore, we observe that the identified parameters of the ConvRNN model reflect the elevation of interior observed in the experiment, while the linear model does not (Fig. 14). Note that the linear model in both cases has identical volume with the printed part only that it fails to capture the inner elevation. We use the identified parameters to predict the height profiles for a four-layered T-shaped frustum and four-layer L-shaped part in Fig. 15, both on an 8 mm $\times$ 8 mm base. The shapes were printed with the same droplet and LS. The ConvRNN prediction yields an rms error of 16 μm for both parts. This corresponds to an accuracy improvement of 54.3% and 44.8% over the prediction of the linear model. It is remarkable that percentage increases in accuracy are comparable to that obtained on self-validation. The cross-sectional views in Fig. 16 show that the convRNN model captures the interior elevations observed in the actual parts. The linear model does not.

2) *Difference in Print/Cure Cycle:* Using only one data point for learning, we examine that the parameters learned in the extreme case of Fig. 11(b) are geometrically independent. For training, we use a four-layered (16 sublayers) cross-shaped part. On self-validation, the identified parameters yield an rms error of 9.7 μm on the printed frustum, an improvement in accuracy of 47.7% over the prediction of the linear dynamic model with the second printing case (Fig. 17). Furthermore, we observe that the identified parameters of the constrained-flow model reflect the contraction of the interior observed in the experiment, while the linear model does not [Fig. 19(a)]. Note that the linear model in both cases has identical volume

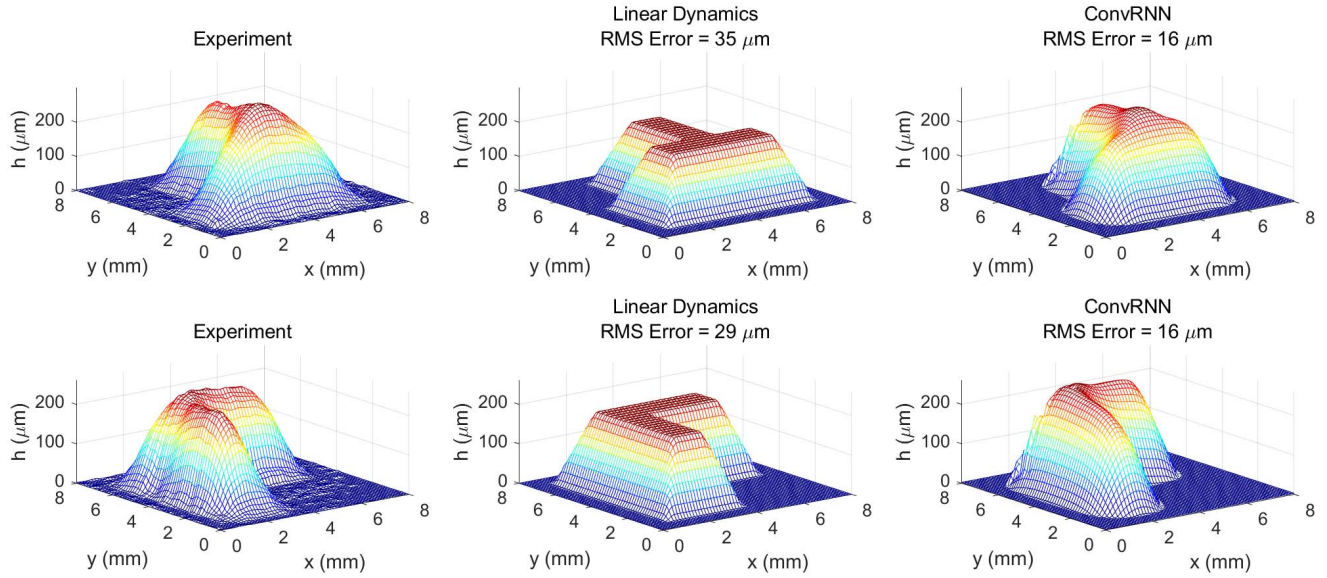


Fig. 15. Performance of proposed model on test parts printed with reduced DS [DS II in Fig. 11(a)]. Observe how the prediction of the proposed model significantly reduces the rms error.

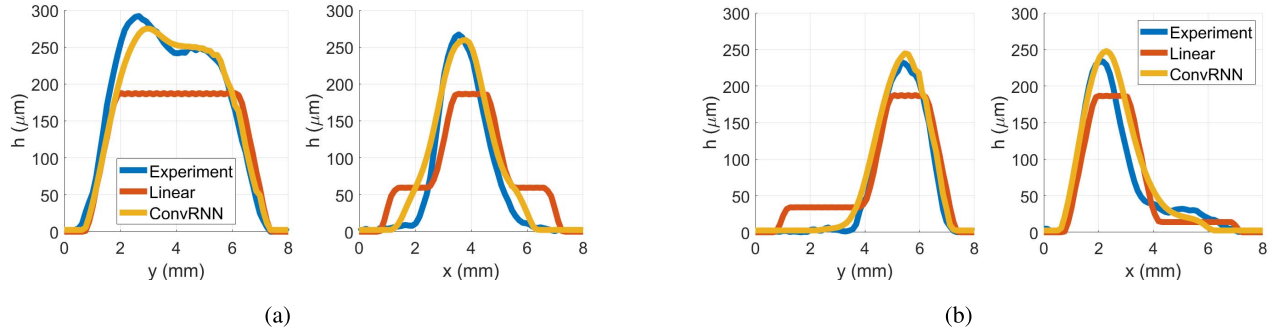


Fig. 16. Sections of (a) T-shaped and (b) L-shaped part used for testing, both printed with reduced DS. Middle cross sections in both x- and y-directions.

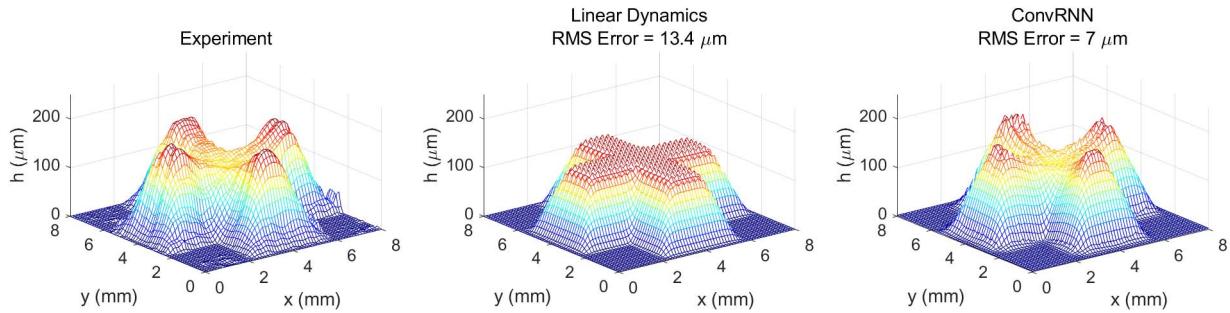


Fig. 17. Comparison between the linear dynamic model and the proposed model for cross-shaped part used for training under the printing scenario in Fig. 11(b).

with the printed part only that it fails to capture the inner elevation.

The identified parameters are used to predict the height profiles for a four-layered T-shaped frustum (Fig. 18). A marked improvement over the prediction of the linear model of up to 30.9% is observed. The cross-sectional views in Fig. 19(b) show that the convRNN model prediction captures the edge elevations observed in the actual part.

C. Discussion

Many machine learning architectures for (especially nonlinear) 2-D or 3-D dynamic systems lack geometry (or boundary) generalizability [39]. This implies that trained model parameters will not predict behavior well when tested on unseen data having a different geometry or boundary. The foregoing results in this section indicate that the model parameters here generalize well across geometries suggesting that the

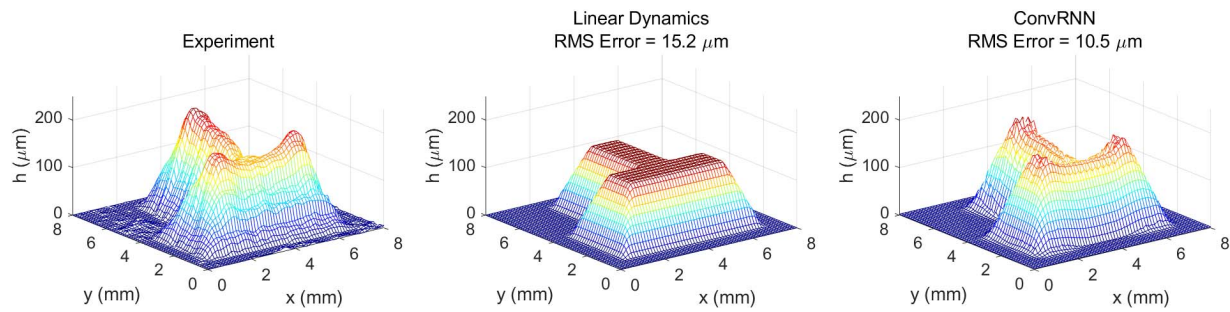


Fig. 18. Comparison between the linear dynamic model and the proposed model for T-shaped part used for testing under the printing scenario in Fig. 11(b).

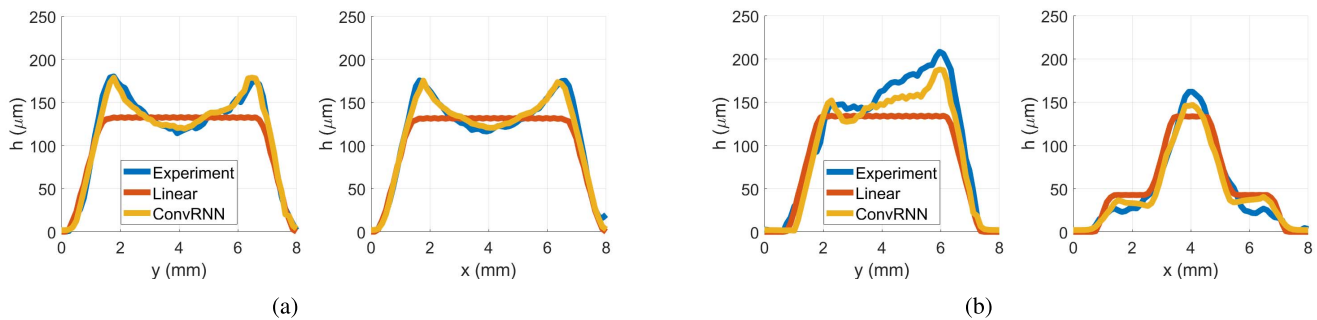


Fig. 19. Section of (a) cross-shaped part used for training (Fig. 15) and (b) T-shaped part for testing (Fig. 14), both printed under the printing scenario in Fig. 11(b).

parameters learned are largely invariant to the geometries used for training. This geometry invariance is enabled by the following:

- 1) learning a 2-D input convolution kernel rather than an unstructured input matrix;
- 2) using a graph structure over the entire printing domain and applying activation functions directly over the link this graph.
- 3) the output activation function for the entire print domain that ensures nonnegativity.

VII. CONCLUSION AND FUTURE WORK

We have developed a conv-RNN model structure for capturing the height evolution during a droplet-based AM process, inkjet 3-D printing, based on physical insight into the printing process. We established the conditions for stability of training the model and how such conditions translate to the physical stability of the height evolution. We experimentally validated the model and showed it to outperform a trained MLP with much less data. The model was also examined under contrasting printing scenarios and proved to require only a little data to learn geometry generalizable parameters in all scenarios. These results not only suggest that the model lends itself to in-process learning (i.e., online refinement as printing data grow), but the structure (or architecture) used here may be extended to multimaterial inkjet 3-D printing and other droplet-based AM processes.

Experiment has shown that the proposed model requires only as much data for training as would be collected in the first few layers of a printing session to be geometrically

independent. This implies that the model may be pretrained on a part's geometry and used to predict and control the height map evolution for another geometry. However, this also suggests that the model structure is amenable for update during the printing process. Hence, future work will aim to demonstrate online learning and geometry-level control with the inkjet 3-D printing process. This calls for an efficient adaptive control scheme with the proposed model structure in which feedback is used to update the model parameters and find suitable input pattern for subsequent layer(s) as the printing progresses [24].

REFERENCES

- [1] S. F. S. Shirazi *et al.*, "A review on powder-based additive manufacturing for tissue engineering: Selective laser sintering and inkjet 3D printing," *Sci. Technol. Adv. Mater.*, vol. 16, no. 3, Jun. 2015, Art. no. 033502.
- [2] M. Singh, H. M. Haverinen, P. Dhagat, and G. E. Jabbour, "Inkjet printing—Process and its applications," *Adv. Mater.*, vol. 22, no. 6, pp. 673–685, 2010.
- [3] Y. Guo, H. S. Patanwala, B. Bogner, and A. W. K. Ma, "Inkjet and inkjet-based 3D printing: Connecting fluid properties and printing performance," *Rapid Prototyping J.*, vol. 23, no. 3, pp. 562–576, Apr. 2017.
- [4] T. Wang, T.-H. Kwok, and C. Zhou, "In-situ, droplet inspection and control system for liquid metal jet 3D printing process," *Proc. Manuf.*, vol. 10, pp. 968–981, Jan. 2017.
- [5] I. A. Spiegel, P. Sammons, and K. Barton, "Hybrid modeling of electrohydrodynamic jet printing," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2322–2335, Nov. 2020.
- [6] Z. Du, R. Xing, X. Cao, X. Yu, and Y. Han, "Symmetric and uniform coalescence of ink-jetting printed polyfluorene ink drops by controlling the droplet spacing distance and ink surface tension/viscosity ratio," *Polymer*, vol. 115, pp. 45–51, Apr. 2017.
- [7] L. Luo, X.-P. Wang, and X.-C. Cai, "An efficient finite element method for simulation of droplet spreading on a topologically rough surface," *J. Comput. Phys.*, vol. 349, pp. 233–252, Nov. 2017.

- [8] C. Doumanidis and E. Skordeli, "Distributed-parameter modeling for geometry control of manufacturing processes with material deposition," *J. Dyn. Syst., Meas., Control*, vol. 122, no. 1, pp. 71–77, Mar. 2000.
- [9] W. Zhou, "Interface dynamics in inkjet deposition," Ph.D. dissertation, School Mech. Eng., Georgia Inst. Technol., Atlanta, GA, USA, 2014.
- [10] A. B. Thompson, C. R. Tipton, A. Juel, A. L. Hazel, and M. Dowling, "Sequential deposition of overlapping droplets to form a liquid line," *J. Fluid Mech.*, vol. 761, pp. 261–281, Dec. 2014.
- [11] Y. Wu and G. Chiu, "Modeling height profile for drop-on-demand print of uv curable ink," in *Proc. ASME Dyn. Syst. Control Conf.*, vol. 2, Oct. 2019, Art. no. V002T13A006.
- [12] X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li, "Applying neural-network-based machine learning to additive manufacturing: Current applications, challenges, and future perspectives," *Engineering*, vol. 5, no. 4, pp. 721–729, Aug. 2019.
- [13] H. Chen and Y. F. Zhao, "Learning algorithm based modeling and process parameters recommendation system for binder jetting additive manufacturing process," in *Proc. 35th Comput. Inf. Eng. Conf.*, Aug. 2015, Art. no. V01AT02A029.
- [14] K. Patan, "Neural network-based model predictive control: Fault tolerance and stability," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1147–1155, Sep. 2015.
- [15] L. H. Gilpin *et al.*, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2018, pp. 80–89.
- [16] Y. Guo and S. Mishra, "A predictive control algorithm for layer-to-layer ink-jet 3D printing," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 833–838.
- [17] D. J. Hoelzle and K. L. Barton, "On spatial iterative learning control via 2-D convolution: Stability analysis and computational efficiency," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1504–1512, Jul. 2016.
- [18] Z. Wang, C. P. Pannier, K. Barton, and D. J. Hoelzle, "Application of robust monotonically convergent spatial iterative learning control to microscale additive manufacturing," *Mechatronics*, vol. 56, pp. 157–165, Dec. 2018.
- [19] C. Pannier, M. Wu, D. Hoelzle, and K. Barton, "LPV models for jet-printed heightmap control," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 5402–5407.
- [20] Y. Guo, J. Peters, T. Oomen, and S. Mishra, "Control-oriented models for ink-jet 3D printing," *Mechatronics*, vol. 56, pp. 211–219, Dec. 2018.
- [21] A. Karpatne *et al.*, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2318–2331, Jun. 2017.
- [22] A. Daw, A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided neural networks (PGNN): An application in lake temperature modeling," 2017, *arXiv:1710.11431*.
- [23] X. Jia *et al.*, "Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles," in *Proc. SIAM Int. Conf. Data Mining*, 2019, pp. 558–566.
- [24] U. Inyang-Udoh and S. Mishra, "A learning-based approach to modeling and control of inkjet 3D printing," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 460–466.
- [25] U. Inyang-Udoh, Y. Guo, J. Peters, T. Oomen, and S. Mishra, "Layer-to-layer predictive control of inkjet 3-D printing," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 4, pp. 1783–1793, Jun. 2020.
- [26] L. Lu, J. Zheng, and S. Mishra, "A layer-to-layer model and feedback control of ink-jet 3-D printing," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1056–1068, Jun. 2015.
- [27] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.
- [28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: <https://www.deeplearningbook.org>
- [29] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Cambridge, MA, USA: MIT Press, 2015, pp. 802–810.
- [30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.
- [31] Z. Allen-Zhu, Y. Li, and Z. Song, "On the convergence rate of training recurrent neural networks," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2019, pp. 6676–6688, Art. no. 599.
- [32] H. Wenbin, L. Y. Tsui, and G. Haiqing, "A study of the staircase effect induced by material shrinkage in rapid prototyping," *Rapid Prototyping J.*, vol. 11, no. 2, pp. 82–89, 2005.
- [33] K. C. Das, "An improved upper bound for Laplacian graph eigenvalues," *Linear Algebra its Appl.*, vol. 368, pp. 269–278, Jul. 2003.
- [34] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Math. Program.*, vol. 89, no. 1, pp. 149–185, 2000.
- [35] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, vol. 28, 013, pp. III-1310–III-1318.
- [36] L. Lu, Y. Shin, Y. Su, and G. Em Karniadakis, "Dying ReLU and initialization: Theory and numerical examples," 2019, *arXiv:1903.06733*.
- [37] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000.
- [38] S. Walczak and N. Cerpa, "Heuristic principles for the design of artificial neural networks," *Inf. Softw. Technol.*, vol. 41, no. 2, pp. 107–117, Jan. 1999.
- [39] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annu. Rev. Fluid Mech.*, vol. 52, no. 1, pp. 477–508, Jan. 2020.



Uduak Inyang-Udoh (Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of Lagos, Lagos, Nigeria, in 2016. He is currently pursuing the Ph.D. degree in mechanical engineering with the Rensselaer Polytechnic Institute (RPI), Troy, NY, USA.

His current research interests include data-driven modeling and control, with application in additive manufacturing systems.



Sandipan Mishra (Member, IEEE) received the B.Tech. degree in mechanical engineering from the IIT Madras, Chennai, India, in 2002, and the Ph.D. degree in mechanical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2008.

He joined the Rensselaer Polytechnic Institute, Troy, NY, USA, as a Faculty Member, in 2010, where he is currently an Associate Professor with the Department of Mechanical, Aerospace, and Nuclear Engineering. His current research interests include

general area of systems and control theory, iterative learning control, optimal control, and precision mechatronics, as applied to autonomous aerial vehicles, additive manufacturing, and smart building systems.