

Layer-to-Layer Predictive Control of Inkjet 3-D Printing

Uduak Inyang-Udoh , Yijie Guo, Joost Peters, Tom Oomen , and Sandipan Mishra 

Abstract—This article develops and experimentally validates a distributed predictive control algorithm for closed-loop control of inkjet 3-D printing to handle constraints, e.g., droplet volume bounds, as well as the large-scale nature of the 3-D printing problem. The large number of decision variables, i.e., droplet volumes at each grid point, in high resolution inkjet 3-D printing makes centralized methods extremely time-consuming, thus, a distributed implementation of the controller is necessary. First, a graph-based height evolution model that captures the liquid spreading dynamics is described. Based on this model, a scalable closed-loop control algorithm using distributed model predictive control (MPC) that can reduce computation time significantly is designed and experimentally implemented. The performance and efficiency of the algorithm are shown to outperform open-loop printing and closed-loop printing with existing centralized MPC methods.

Index Terms—Additive manufacturing (AM), inkjet 3-D printing, model predictive control (MPC).

I. INTRODUCTION

INKJET 3-D printing is a type of additive manufacturing (AM) process, which builds 3-D parts layer by layer. In inkjet 3-D printing, photocurable plastic resins are directly deposited by ejection from a nozzle to build 3-D parts, with ultraviolet (UV) light curing in between layers. Several commercial printers build parts using this process (PolyJet printers by Stratasys and MultiJet printers by 3-D Systems).

The need for closed-loop control has been well-recognized in the AM literature. There have been significant advances in geometry-level feedback control in many AM processes (such as laser metal deposition [1], [2], laser metal-wire deposition [3],

laser cladding [4], and exposure-controlled projection lithography [5]), where height measurement feedback has been used for model-based control of layer geometry. For inkjet 3-D printing, real-time control of process parameters such as building plate temperature, nozzle back pressure [6], and jetting frequency [7] has already been demonstrated in prior literature. However, related work on geometry level in-process feedback control is sparse. The number of layers to be deposited and droplet patterns for each layer are determined in advance and are not adjusted through feedback. This open-loop printing strategy makes the printing process vulnerable to process uncertainties such as droplet shapes, sizes and locations, resulting in parts that have poor geometric accuracy.

Early work on geometry-level control of 3-D printing started from partial (local) geometry control, for example, to regulate the droplet shape [8], or on a higher level, to control the bead width and height [9]. More recently, closed-loop control methods that directly monitor the whole-part geometry have been proposed. In [10], a greedy geometric feedback algorithm that iteratively searches for locations to deposit droplets based on the tracking error of geometric shape is proposed and demonstrated. Naturally, this greedy search approach cannot account for interactions between droplets and the surface evenness since it does not model these effects.

In other more recent work, each droplet deposition is modeled as a spatial impulse response or 2-D convolution kernel [11]. The height evolution is then assumed to be a superposition of the droplets [12]–[14]. While the model parameters in [11]–[13] are linear time-invariant (LTI), those in [14] vary spatially and in time. This enables [14] to capture nonlinearities associated with deposition and the effect of the surface below. However, as with the LTI models, it assumes no neighboring droplet interactions. Nonetheless, control algorithms based on such LTI models have been developed and shown to improve geometry tracking. Lu *et al.* [15] present an empirical model that incorporates the effect of neighboring droplets. The predictive control algorithm based on this model can address both geometry tracking and surface evenness; but because of the nonlinearity of the model, it suffers from poor scalability for high-resolution printing. Even with a linear model, as the size of the printing region increases, the number of grid points increases exponentially and the control problem becomes computationally burdensome.

To address this issue, a distributed model predictive control (MPC) scheme for inkjet 3-D printing was proposed in [16]. Here, a linear model that incorporates interaction between droplets was used. It was shown in simulation that in comparison

Manuscript received January 8, 2020; revised April 3, 2020; accepted April 27, 2020. Date of publication June 4, 2020; date of current version August 13, 2020. This work was supported in part by the National Science Foundation (NSF) Career Award grant CMMI-1254313 and the NSF Data-Driven Cyberphysical Systems Award grant CNS-1645648. Recommended by Technical Editor J. Yi and Senior Editor X. Chen. (Corresponding author: Uduak Inyang-Udoh.)

Uduak Inyang-Udoh and Sandipan Mishra are with the Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: inyanu@rpi.edu; mishrs2@rpi.edu).

Yijie Guo is with UBTECH Robotics Beijing Branch, Beijing 100083, P.R.China (e-mail: fugo.guo@ubtrobot.com).

Joost Peters is with TNO Technical Sciences, 2600 AD Delft, The Netherlands (e-mail: joost.peters@tno.nl).

Tom Oomen is with the Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: T.A.E.Oomen@tue.nl).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2020.2999873

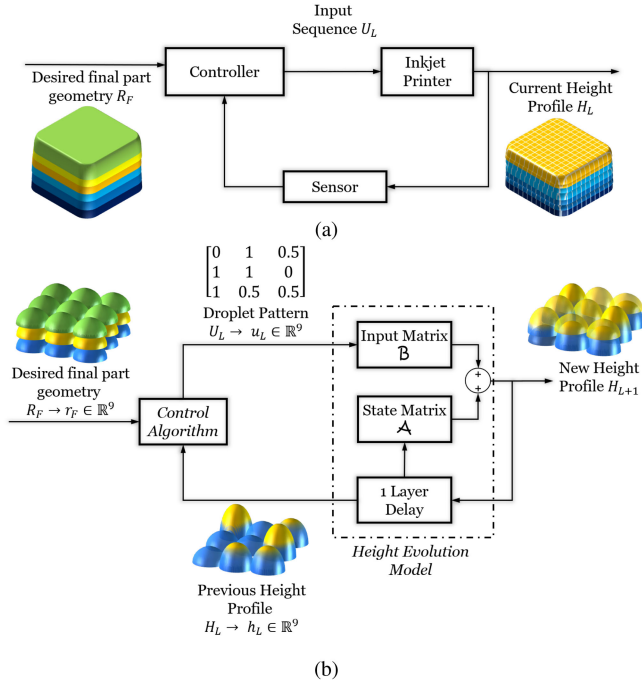


Fig. 1. (a) Schematic block diagram of the closed-loop layer-to-layer printing process. (b) Illustration of closed-loop printing process with layer-to-layer height evolution model.

to centralized MPC, the distributed controller offers significant reduction in computational time without compromise in tracking accuracy. This article extends the work presented in [16] through experimental demonstration and validation of the MPC scheme. Furthermore, we prove analytically that the proposed MPC algorithm stabilizes the closed-loop system. The computational and tracking performance of the control algorithm is evaluated by printing various geometries. We show that the closed-loop scheme provides significant performance improvement over open-loop printing, while being computationally scalable.

The rest of this article is organized as follows. The general printing control problem is described in Section II. Next, the model used in the control algorithm design is presented in Section III. The MPC problem is formulated in Section IV, while the distributed MPC-based algorithm is developed in Section V. Then, in Section VII, experimental printing results using the proposed algorithm is compared with open-loop printing. Finally, Section VIII concludes this article.

II. PROBLEM DESCRIPTION

The general formulation of the printing control problem is presented in this section. The printing region is discretized into an $n_x \times n_y$ grid space based on the printing resolution: total number of points $n = n_x \cdot n_y$. After printing each layer, the height profile of the printed part is measured and the controller uses the measurement to generate an appropriate grid pattern of droplets for the next layer. This closed-loop layer-to-layer printing process is as shown in Fig. 1(a). R_F , H_L , and $U_L \in \mathbb{R}^{n_x \times n_y}$ are matrices denoting the final height profile, the

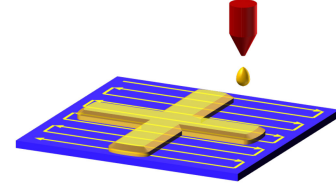


Fig. 2. Printing process for one layer. The nozzle moves along a predetermined path and deposits droplets sequentially.

current height profile, and the input sequence (droplet pattern) at layer L .

For effective geometric control of the shape of the part, a model that captures the printing dynamics is necessary. To model the dynamics, we express R_F , H_L , and U_L in vectorized form r_F , h_L , and $u_L \in \mathbb{R}^n$. Fig. 1(b) illustrates the printing process for a 3×3 grid size and the printer replaced with a height-evolution model (presented in Section III). The control objective is to generate a droplet pattern u_L that tracks a reference geometry r_L for each layer. The input droplet volume deposited at each grid location i is constrained by: $u_L(i) \in [V_{\min}, V_{\max}]$, where $V_{\min}$ and $V_{\max}$ are the minimum and maximum droplet volumes. Specifically, the goal is to minimize the geometric tracking error $e_{L+1} = h_{L+1} - r_{L+1}$ based on feedback of the height profile h_L .

III. MODEL DESCRIPTION

In this section, the height evolution model to capture deposition and spreading of the ink droplets in [17] [see Fig. 1(b)] is presented. The key idea is that fluid flows from higher to lower heights proportional to the height differences.

A. Layer-to-Layer Height Evolution Model

During the printing process, droplets are sequentially deposited along a predetermined printing path, which usually is a raster path, as shown in Fig. 2. It is shown in [17] that by using a directed graph, the height evolution from time step to time step can be written as

$$h_{k+1} = A_k h_k + B_k u_k \quad (1)$$

where $h_k \in \mathbb{R}^n$ is the column vector that represents the height profile at the k th time step; $A_k = (I - DF_k D^T)$ is the state matrix that captures the effect of liquid flow between neighboring locations, $D \in \mathbb{R}^{n \times l}$ being the incidence matrix corresponding to a digraph with l links, and $F \in \mathbb{R}^{l \times l}$, a diagonal positive definite matrix containing the flowability parameters; and, $B_k \in \mathbb{R}^n$ is the vector containing a unit droplet's shape and location at the k th time step.

Moreover, when the printing trajectory is repeated, the time-step height evolution of (1) can be lifted for each layer to yield a layer-to-layer height evolution model

$$h_{(L+1) \cdot n} = \mathcal{A} h_{L \cdot n} + \mathcal{B} \begin{bmatrix} u_1 & u_2 & \dots & u_k & \dots & u_n \end{bmatrix}^T \quad (2)$$

where $\mathcal{A} \in \mathbb{R}^{n \times n}$ is $\prod_{i=n}^1 A_i$ and $\mathcal{B} \in \mathbb{R}^{n \times n}$ is

$$\left[\left(\prod_{i=n}^2 A_i \right) B_1 \left(\prod_{i=n}^3 A_i \right) B_2 \cdots \left(\prod_{i=n}^{k+1} A_i \right) B_k \right. \\ \left. \left(\prod_{i=n}^{k+2} A_i \right) B_{k+1} \cdots A_n B_{n-1} B_n \right].$$

Henceforth, for brevity, we abuse notation and denote $h_{L,n} \equiv h_L$, the height profile of the L th layer.

B. Model Properties

Since this model is based on fluid flow from higher to lower heights and volume conservation, the open-loop system is stable at each time step. Note $(\rho(A_k) \leq 1 \forall k)$ and furthermore, since A_k is symmetric, $\sigma_{\max}(A_k) = \rho(A_k) \leq 1 \forall k$. As a result, $\mathcal{A} = \prod_{i=n}^1 A_i$ satisfies:

- 1) $\mathcal{A}$ is Schur, that is, $\rho(\mathcal{A}) \leq 1$ where $\rho(\mathcal{A})$ is the spectral radius of $\mathcal{A}$;
- 2) the layer-to-layer height evolution is passive, that is, $\sigma_{\max}(\mathcal{A}) \leq 1$.

Proof: Since $\sigma_{\max}(A_k) \leq 1 \forall k$, $\sigma_{\max}(\mathcal{A}) \leq \prod_{i=n}^1 \sigma_{\max}(A_i) \leq 1$.

Furthermore, $\rho(\mathcal{A}) \leq \sigma_{\max}(\mathcal{A})$. Hence, both statements mentioned above hold true. ■

IV. MODEL PREDICTIVE CONTROL

In this section, a model predictive controller for the inkjet printer is formulated. Several features of the printing process make a predictive controller suitable for the geometry-level control: the reference profiles of all future layers are known *a priori*; the profile of each layer can be measured after it is printed; the reference profile and control space are high dimensional; and the control input is constrained.

A. Centralized MPC

Suppose we wish to find a sequence of optimal control inputs that track the reference over the next N layers. This may be formulated as a standard MPC problem, where a cost-function is minimized over a finite receding horizon of N layers

$$\min_{U_L} J(H_L, U_L) \\ \text{s.t. } h_{i+1|L} = \mathcal{A}h_{i|L} + \mathcal{B}u_{i|L}, \quad i = \{0, 1, \dots, N-1\} \quad (3) \\ \mathcal{E}U_L^T \leq c$$

where $U_L = [u_{0|L}, \dots, u_{N-1|L}]^T$, $u_{i|L}$ is the i th layer control in the receding horizon, $u_{0|L} = u_L$, while $H_L = [h_{1|L}, \dots, h_{N|L}]^T$, $h_{i|L}$ is the i th layer height profile in the same horizon, with $h_{0|L} = h_L$ being the current measurement.

The cost function is designed to penalize tracking error

$$J(H_L, U_L) = (h_{N|L} - r_{N|L})^T P (h_{N|L} - r_{N|L}) \\ + \sum_{i=0}^{N-1} [(h_{i+1|L} - r_{i+1|L})^T Q (h_{i+1|L} - r_{i+1|L}) + u_{i|L}^T R u_{i|L}] \quad (4)$$

where $r_{i|L}$ is the corresponding i th reference profile in the receding horizon at layer L . P , Q , and $R \in \mathbb{R}^{n \times n}$ are (semi)

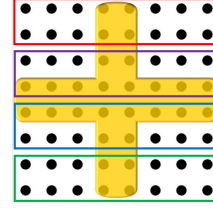


Fig. 3. To decompose the optimization problem, the printing region is separated into l subregions (here, four subregions).

positive-definite matrices; Q and R are the state and input cost matrices, respectively, and P is the terminal cost matrix that can be designed to guarantee MPC stability. In (3), $\mathcal{E}$ and c are, respectively, a matrix and a column vector defined by

$$\mathcal{E} = \begin{bmatrix} E_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & E_{N-1} \end{bmatrix} \quad c = \begin{bmatrix} b_0 \\ \vdots \\ b_{N-1} \end{bmatrix} \quad (5)$$

where $E_i = [-I \quad I]^T$, and $b_i = [-u_{\text{low}} \quad u_{\text{high}}]^T$ defines the upper and lower constraints on the input. The optimization is performed each layer, and $u_{0|L}^*$ is applied.

This MPC problem can be solved using well-known (centralized) MPC design methods for quadratic convex problems [18]. However, for increasing size of the optimization problem (prediction horizon N , grid size n), computation times using standard MPC can become unfeasible. Centralized formulations, scale with $O(N^3 n^3)$ using, for example, interior-point convex solvers [19]. In the following section, a distributed approach to the problem that reduces computation time is developed.

V. DISTRIBUTED MPC

In this section, we present a distributed control algorithm that provides efficient solution to the large predictive control that was presented in [12]. First, the centralized control problem is partitioned with the printing region. Then, the dual decomposition method is used for the distributed algorithm. Finally, we certify stability of the closed-loop system.

A. Partitioning of the Optimization Problem

The whole printing region is separated into l subregions, as shown in Fig. 3. Consider the partitioning of the layer evolution model (2) introduced in Section III

$$\begin{bmatrix} h_{L+1}^1 \\ h_{L+1}^2 \\ \vdots \\ h_{L+1}^l \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} & \cdots & \mathcal{A}_{1l} \\ \mathcal{A}_{21} & \mathcal{A}_{22} & \cdots & \mathcal{A}_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_{l1} & \mathcal{A}_{l2} & \cdots & \mathcal{A}_{ll} \end{bmatrix} \begin{bmatrix} h_L^1 \\ h_L^2 \\ \vdots \\ h_L^l \end{bmatrix} \\ + \begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} & \cdots & \mathcal{B}_{1l} \\ \mathcal{B}_{21} & \mathcal{B}_{22} & \cdots & \mathcal{B}_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_{l1} & \mathcal{B}_{l2} & \cdots & \mathcal{B}_{ll} \end{bmatrix} \begin{bmatrix} u_L^1 \\ u_L^2 \\ \vdots \\ u_L^l \end{bmatrix}. \quad (6)$$

For the partitioned system (6), the cost function becomes

$$J(h_L, U_L) = \sum_{j=1}^l ((h_{N|L}^j - r_{N|L}^j)^T P_j (h_{N|L}^j - r_{N|L}^j) + \sum_{i=0}^{N-1} [(h_{i|L}^j - r_{i|L}^j)^T Q_j (h_{i|L}^j - r_{i|L}^j) + (u_{i|L}^j)^T R_j (u_{i|L}^j)]). \quad (7)$$

Thus, a partitioned quadratic program (QP) is constructed

$$\begin{aligned} & \underset{x}{\text{minimize}} && x^T H x \\ & \text{subject to} && Fx = z \\ & && Tx \leq q \end{aligned} \quad (8)$$

where the optimization variable consists of both the predicted tracking error and future inputs

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \end{bmatrix}, \quad x_j = \begin{bmatrix} h_{0|L}^j - r_{0|L}^j \\ h_{1|L}^j - r_{1|L}^j \\ \vdots \\ h_{N|L}^j - r_{N|L}^j \\ u_{0|L}^j \\ u_{1|L}^j \\ \vdots \\ u_{N-1|L}^j \end{bmatrix} \quad \forall j \in \{1, \dots, l\}. \quad (9)$$

The Hessian H consists of the tracking penalty matrices. It is a diagonal matrix that is described by

$$H = \begin{bmatrix} H_{11} & 0 & 0 & 0 \\ 0 & H_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & H_{ll} \end{bmatrix}$$

$$H_{jj} = \begin{bmatrix} Q_j & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_j & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P_j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_j \end{bmatrix} \quad j \in \{1, \dots, l\}.$$

The equality constraint captures the layer dynamics and can be partitioned similarly

$$\begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1l} \\ F_{21} & F_{22} & \cdots & F_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ F_{l1} & F_{l2} & \cdots & F_{ll} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_l \end{bmatrix} \quad (10)$$

where

$$F_{ii} = \begin{bmatrix} -I & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \mathcal{A}_{ii} & -I & \cdots & 0 & 0 & \mathcal{B}_{ii} & 0 & \cdots & 0 \\ 0 & \mathcal{A}_{ii} & \cdots & 0 & 0 & 0 & \mathcal{B}_{ii} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{A}_{ii} & -I & 0 & 0 & \cdots & \mathcal{B}_{ii} \end{bmatrix} \quad (11)$$

$$F_{ij} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \mathcal{A}_{ij} & 0 & \cdots & 0 & 0 & \mathcal{B}_{ij} & 0 & \cdots & 0 \\ 0 & \mathcal{A}_{ij} & \cdots & 0 & 0 & 0 & \mathcal{B}_{ij} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{A}_{ij} & 0 & 0 & 0 & \cdots & \mathcal{B}_{ij} \end{bmatrix}. \quad (12)$$

The column vector z is defined as

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_l \end{bmatrix} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_l \end{bmatrix} - F \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_l \end{bmatrix} \quad (13)$$

where the elements in z are composed of

$$W_j = \begin{bmatrix} h_{0|L}^j \\ 0_1 \\ 0_2 \\ \vdots \\ 0_N \end{bmatrix}, \quad \Gamma_j = \begin{bmatrix} r_{0|L}^j \\ r_{1|L}^j \\ \vdots \\ r_{N|L}^j \\ 0_0 \\ 0_1 \\ \vdots \\ 0_{N-1} \end{bmatrix} \quad \forall j \in \{1, \dots, l\}. \quad (14)$$

The inequality constraint in the QP (8) is decomposed similarly

$$\begin{bmatrix} T_{11} & 0 & 0 & 0 \\ 0 & T_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & T_{ll} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \end{bmatrix} \leq \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_l \end{bmatrix} \quad (15)$$

where T_{jj} and q_j are described by

$$T_{jj} = \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{E}_j \end{bmatrix}, \quad q_j = \begin{bmatrix} 0 \\ c_j \end{bmatrix} \quad \forall j \in \{1, \dots, l\}. \quad (16)$$

In this formulation, x , H , F , z , T , and q remain separable into l building blocks such that the problem is naturally decomposable. For a large optimization problem that is coupled, dual decomposition allows for separation of the minimization-step of the central problem [20]. This method solves the optimization problem by price discovery using a gradient step. For a set price, the problem becomes separable and can be solved in parallel, which allows for distributed optimization.

B. Distributed MPC

This subsection provides the algorithm for distributed MPC for the height tracking problem. The algorithm uses the dual decomposition method discussed in [20] to solve the decomposable optimization problem of Section V-A. For the partitioned optimization problem in (8), we construct the Lagrangian

$$L(x, \lambda) = \sum_{j=1}^l \left[x_j^T H_j x_j + \lambda_j^T \left(z_j - \sum_p [F_{jp} x_p] \right) \right]. \quad (17)$$

The decomposed optimization problem is now given by [21]

$$\begin{aligned} \max_{\lambda} \quad & \min_x \sum_{j=1}^l \left[x_j^T H_j x_j + \lambda_j^T \left(z_j - \sum_p [F_{jp} x_p] \right) \right] \\ \text{subject to} \quad & Tx \leq q. \end{aligned} \quad (18)$$

This optimization problem may be rewritten as

$$\begin{aligned} \max_{\lambda} \quad & \sum_{j=1}^l \left[\min_{x_j} \underbrace{x_j^T H_j x_j + \lambda_j^T z_j - x_j^T \sum_p [F_{jp} \lambda_p]}_{\tilde{L}_j} \right] \\ \text{subject to} \quad & T_{jj} x_j \leq q_j. \end{aligned} \quad (19)$$

Notice that the new problem involves maximizing a sum of l separate and smaller QPs $\tilde{L}_j$ over “price” λ . When the maximum is obtained, the constraints are, by definition, satisfied. The optimal price is found by gradient ascent with the “price update”:

$$\lambda^{s+1} = \lambda^s + \gamma^s \nabla g^s(\lambda) \quad (20)$$

where γ^s is the step size. We compute the step size using an approximate Newton’s method given in [22], avoiding the expensive computation of the Jacobian ∇g

$$\gamma^s = \frac{-(\nabla g^s - \nabla g^{s-1})^T (\lambda^s - \lambda^{s-1})}{(\nabla g^s - \nabla g^{s-1})^T (\nabla g^s - \nabla g^{s-1})}. \quad (21)$$

With the defined local minimization in (19) and price update method in (20) and (21), an algorithm for distributed MPC can be constructed. The algorithm is summarized by the pseudo code presented in Algorithm 1 and presented briefly in the following:

- 1) In lines 1–4, the input and initialization of the algorithm is defined. The input includes the current height profile h_L , reference profile r_L , (if applicable) the optimal price of the previous layer $\lambda_{L-1}^{S_k}$, the MPC problem matrices (H, F, T, q) , the number of partitions l and the price convergence criterion Λ .
- 2) In lines 5–7, the local minimization is performed based on the current price λ^s .
- 3) In lines 8–10, the price update is performed by a gradient ascent step.
- 4) In lines 11–16, the stopping condition is posed based on convergence of the price λ . After convergence, the optimal input for the next layer is extracted.

Algorithm 1: Distributed MPC Algorithm.

```

1: Input:  $h_L, r_L, \lambda_{L-1}^{S_k}, l, (H, F, T, q), s_{\max}, \Lambda_\lambda$ 
2: Output:  $u_L$ ,
3: Initialize:  $\gamma^{(0)}, \lambda_L^{(0)} = \lambda_{L-1}^{S_k}$ 
4: for  $s = 1, 2, \dots, s_{\max}$  do
5:   for  $j = 1, 2, \dots, l$  do
6:     Solve  $x_j^{(s+1)} = \underset{x_j}{\operatorname{argmin}} \underbrace{x_j^T H_j x_j + (\lambda_j^{(s)})^T z_j - x_j^T \sum_p [F_{jp} \lambda_p^{(s)}]}_{\tilde{L}_j(x_j, \lambda^{(s)})}$ 
7:   end for
8:   Compute  $\nabla g^{(s)} = (Fx - z)$ ,
9:    $\gamma^{(s)} = \frac{-(\nabla g^{(s)} - \nabla g^{(s-1)})^T (\lambda^{(s)} - \lambda^{(s-1)})}{(\nabla g^{(s)} - \nabla g^{(s-1)})^T (\nabla g^{(s)} - \nabla g^{(s-1)})}$ 
10:  Update  $\lambda^{(s+1)} = \lambda^{(s)} + \gamma^s \nabla g^{(s)}$ 
11:  if  $\frac{(\|\lambda^{(s+1)}\| - \|\lambda^{(s)}\|)}{\|\lambda^{(s)}\|} \leq \Lambda_{\|\lambda\|}$  then
12:     $S_k = s$ 
13:    Extract  $u_L$  from  $x^{S_k}$ 
14:    Break;
15:  end if
16: end for
    
```

C. Stability of MPC

Stability of MPC algorithms for LTI systems has been well-studied since early 1980s [23], where the value function of a finite horizon optimization problem is used as a Lyapunov function to establish stability. Here, the so-called direct method [24] based on the idea employed in [25] and [26] is applied to show stability. For an MPC with the cost function

$$\begin{aligned} J(h_L, U_L) = & (h_{N|L} - r_{N|L})^T P (h_{N|L} - r_{N|L}) \\ & + \sum_{i=0}^{N-1} [(h_{i|L} - r_{i|L})^T Q (h_{i|L} - r_{i|L}) \\ & + (u_{i|L} - u_{i|L}^*)^T R (u_{i|L} - u_{i|L}^*)] \end{aligned} \quad (22)$$

where $h_{0|L} = h_L$, $u_{i|L}^*$ is the ideal input that satisfies

$$r_{i|L+1} = \mathcal{A} r_{i|L} + \mathcal{B} u_{i|L}^*. \quad (23)$$

Stability Lemma: The closed-loop MPC system

$$h_{L+1} = \mathcal{A} h_L + \mathcal{B} K_{\text{MPC}}(h_L) \quad (24)$$

is asymptotically stable at the point $h_L = r_L, u_L = u_L^*$ if the following conditions are satisfied:

- 1) $Q \succ 0$ and $R \succ 0$;
- 2) the terminal cost weight P satisfies $P \succ 0$ and

$$(\mathcal{A} + \mathcal{B}K)^T P (\mathcal{A} + \mathcal{B}K) - P \preceq -Q - K^T R K \quad (25)$$

where K is any matrix with $\rho(\mathcal{A} + \mathcal{B}K) < 1$.

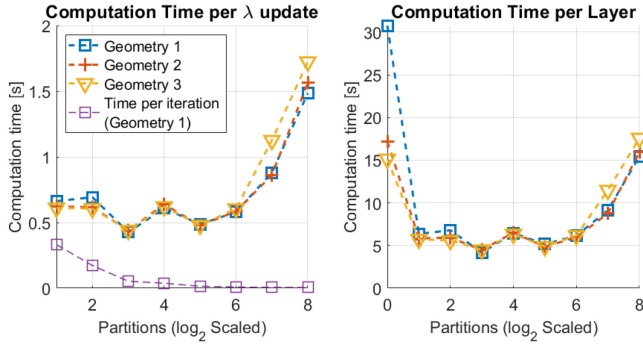


Fig. 4. Algorithm 1 computation time for various number of partitions l . Simulations are run for three 64×64 geometries with $N = 2$. (Left) Average time taken for the local minimization in lines 5–7 of Algorithm 1. (Right) Mean time taken for the solving the entire dual problem at each layer (lines 4–16). Lowest computational times lie in $2 \leq l \leq 2^6$.

Proof: The proof is standard and is in the Appendix. ■

Remark: This proof is contingent on finding a stabilizing feedback control K , such that (25) can be used to obtain P from Q . Since $\mathcal{A}$ is stable, K can be trivially chosen to be $\mathbf{0}$. Furthermore, since $\mathcal{A}$ is passive, K can be any passive feedback controller, thereby expanding the design space of P and Q . Finally, because the open-loop system is passive, the algorithm is inherently robust to small parametric variations.

VI. ALGORITHMIC PERFORMANCE

This section evaluates the algorithmic performance of the distributed MPC against centralized MPC. First, we discuss the effect of partitioning the MPC. Next, we compare performance based on layer height tracking and the computational expense of the distributed and centralized schemes at different grid sizes and prediction horizons.

A. Effect of Partitioning

In Algorithm 1, the number of partitions l influences both how many quadratic subproblems $\tilde{L}_j$ will be solved for a given price update and how many price updates are required for convergence. Increasing the number of partitions increases the number of subproblems but lowers the cost of each. Fig. 4 shows how the number of partitions affect computational time of the distributed algorithm for three different five-layered parts simulated on a $n = n_x \cdot n_y = 64 \times 64$ grid space with prediction horizon $N = 2$. The average time taken for each subproblem decreases with number of partitions. However, the average time for all subproblems only slightly decreases with increase in partitioning; beyond $2^6 = \sqrt{n}$, the computational cost rises.

In general, for a given price λ , it is computationally less expensive to minimize the set of quadratic problems $\{\tilde{L}_j : 1 \leq j \leq l\}$ than the primal problem in (8). However, the minimization of all $\tilde{L}_j$ must be recomputed iteratively till the price λ converges. The right plot in Fig. 4 show the total time taken (averaged across layers) to solve the dual problem till convergence for different partition numbers. Note that the convergence criterion was kept

TABLE I

NUMBER OF ITERATIONS (PRICE UPDATES) AND TRACKING ERROR ON SOLVING ALGORITHM 1 FOR A 64×64 FIVE-LAYERED GEOMETRY AT VARIOUS NUMBER OF PARTITIONS l

l	2	4	8	16	32	64	128	256
Iterations (Average)	9.6	9.8	9.8	10.6	10.6	10.6	10.4	10.4
$\ e\ _2$ Tracking Error (mm)	0.33	0.32	0.32	0.32	0.32	0.32	0.32	0.32

TABLE II

TWO-NORM OF THE LAYER HEIGHT TRACKING $\|e\|_2$ FOR OPEN-LOOP, CENTRALIZED, AND DISTRIBUTED MPC

	OL [mm]	CMPC [mm]			DMPC [mm]		
$\sqrt{n}$	$\ e\ _2$	$\ e\ _2$ (N=1)	$\ e\ _2$ (N=3)	$\ e\ _2$ (N=5)	$\ e\ _2$ (N=1)	$\ e\ _2$ (N=3)	$\ e\ _2$ (N=5)
80	1.14	0.85	0.69	0.67	0.84	0.69	0.68
90	1.22	0.92	0.73	0.71	0.92	0.73	0.72
100	1.31	0.99	0.77	0.75	0.99	0.77	0.76
110	1.41	1.05	0.82	0.79	1.05	0.81	0.80
120	1.51	1.14	0.86	0.84	1.14	0.85	0.84

constant for all partitions. The plot shows similar trend as that on the left suggesting that the number of iterations (recomputations to update price) required for convergence does not vary much. This is in fact shown in Table I. Table I additionally shows that the tracking error is not changed by the number of partitions. In the rest of this work l is kept less than $(N - 1)\sqrt{n}$.

B. Height Tracking Performance

Next, we simulate open-loop and closed-loop MPC printing for 15 layers of a cross-shaped part. At each layer, we evaluate the optimal input for the successive layer(s). In the open-loop simulation, a predetermined input profile of the next layer is used. On the other hand, the centralized MPC is implemented by solving (3) with knowledge of the current height profile and reference profiles of N subsequent layers. The distributed MPC is implemented following Algorithm 1 with $l = (N - 1)\sqrt{n}/10$. For capturing jetting uncertainty, droplet shapes for B_k are selected randomly from a set of experimentally measured droplets. Meanwhile, in the nominal model used for MPC, the average droplet shape is used.

Table II summarizes the open-loop, centralized MPC and distributed MPC performances in terms of layer height tracking for different grid sizes and prediction horizons. The centralized MPC and distributed MPC improve the layer height tracking compared to open-loop printing. We observe that the distributed MPC provides nearly identical tracking performance compared to the centralized MPC, as deviations in terms of layer height tracking are within margins of uncertainty for all simulations conducted. This implies that the solution to the distributed problem is close to the centralized solution.

C. Computational Performance

We now compare the computation times for the centralized and distributed MPC algorithms for different grid sizes and prediction horizons. The computation time results are summarized

TABLE III
AVERAGE COMPUTATION TIME (ONE LAYER) FOR
CENTRALIZED AND DISTRIBUTED MPC

$\sqrt{n}$	CMPC in [s]			DMPC in [s]		
	N=1	N=3	N=5	N=1	N=3	N=5
80	12.4	166.4	461.6	3.1	14.9	27.5
90	15.9	204.8	644.9	3.8	19.5	35.9
100	21.5	280.4	853.7	4.6	24.3	45.9
110	26.9	354.8	1169.8	5.4	28.3	55.6
120	34.7	484.8	1663.4	6.3	33.3	68.3

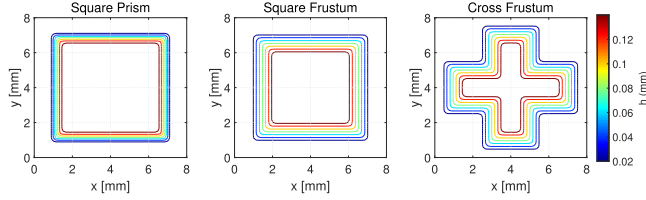


Fig. 5. Reference profiles for three geometries printed.

in Table III for the two methods for comparison. It is evident that the distributed approach significantly reduces the computation time; in most of the cases, by over an order of magnitude. The distributed approach also shows better scaling in terms of grid size: the computation time scales only linearly with the grid size in case of distributed MPC; the scaling is exponential with the centralized MPC.

VII. EXPERIMENTAL RESULTS

In the previous section, it was shown through simulation studies that the distributed algorithm significantly reduces computational time as to make feedback control in real-time practicable. It was shown that this significant reduction in computational time does not compromise layer height tracking accuracy. Hence, it suffices in practice to implement feedback control with the distributed algorithm. In this section, we present results from *experimental implementation* of the distributed controller on several printed parts.

Experiments are conducted on the inkjet 3-D printer setup described in [15]. The setup comprises an ink-jetting system that dispenses the liquid polymer, a motion system to drive the substrate as printing proceeds, an UV curing box that solidifies the liquid material and a 2-D laser sensor, which measures the printed part's height profile. The components are controlled from a computer.

1) *Candidate test parts:* To validate the DMPC algorithm, we demonstrate the performance of the closed-loop algorithm under different experimental scenarios. Three geometries (a square prism, a square frustum, and a cross-shaped frustum) with references shown in the Fig. 5 were printed. The sides of all geometries slope since vertical walls cannot be fabricated with straight edges using an inkjet printer. The first geometry is basic, sloping only slightly as is necessary for fabrication and measurement. The second shape is less prismatic, that is, the cross section of each layer is varying. Finally, for the third case,

TABLE IV
NORMALIZED DIFFERENCE BETWEEN CMPC
AND DMPC CONTROL INPUT (δ_u)

Part Shape	Layer 1	Layer 2	Layer 3
Square prism	0.027	0.037	0.036
Square frustum	0.049	0.057	0.052
Cross frustum	0.060	0.054	0.057

we print a thin and tall cross-shaped part where inward corners (nonconvex geometry) are present.

2) *Algorithm parameters:* The distributed algorithm was initialized with the following parameters: partitioning $l = 4$, receding horizon $N = 2$, grid size $n = n_x \times n_y = 64 \times 64$, maximum number of iteration $S_{\max} = 300$, and price convergence value $\Lambda = 10^{-6}$. The cost function weights are selected as $P = Q = 10^5 R \in \mathbb{R}^{n \times n}$ for control emphasis on height tracking. The algorithm was implemented on a 3.4 GHz Intel Core i7 16 GB SDRAM computer.

3) *Control implementation:* The controlled printing process proceeds as follows. At each layer, the current height profile is measured. With the given reference profile and the abovementioned parameters, Algorithm 1 is solved to find the next layers control input u_L . Since the printing setup accepts only quantized values for u_L , each QP $\tilde{L}_j$ in the algorithm is solved using Gurobi's mixed-integer solver on MATLAB. For comparison, all three geometries are also printed in open loop.

A. Comparison of Control Inputs Produced by Centralized MPC and Distributed MPC

We first establish that the centralized and distributed MPC algorithms effectively produce the same control input. In order to do so, for each of the test parts, we compare the control input produced by the CMPC and the DMPC algorithms *from the same experimental height profile measurement*. We then evaluate them by determining the normalized error between the two

$$\delta_u = \frac{\|u_{\text{CMPC}} - u_{\text{DMPC}}\|_2}{\|u_{\text{CMPC}}\|_2}$$

in Table IV. We note that the worst-case difference is within 6%. Thus, these results demonstrate that the inputs from the CMPC and DMPC are identical to within the precision of what can be commanded to the printer. Hence, we proceed only with the DMPC given the computational burden imposed by the CMPC.

B. Comparison With Open-Loop Control

We now present a comparison of printing accuracy between open-loop and closed-loop distributed MPC. The reference geometry and printing results for the square prism and cross frustum are shown in Figs. 6 and 7, respectively. The error profiles are shown in Figs. 8 and 9. These profiles emphasize the improvement in geometric accuracy. The RMS errors from the open- and closed-loop printings are summarized in Table V. The table also shows the side-wall RMS error of each part, indicating that both side-wall accuracy and overall surface accuracy are significantly improved. RMS errors of $16.4 \mu\text{m}$, $11.3 \mu\text{m}$, and $13.8 \mu\text{m}$ are

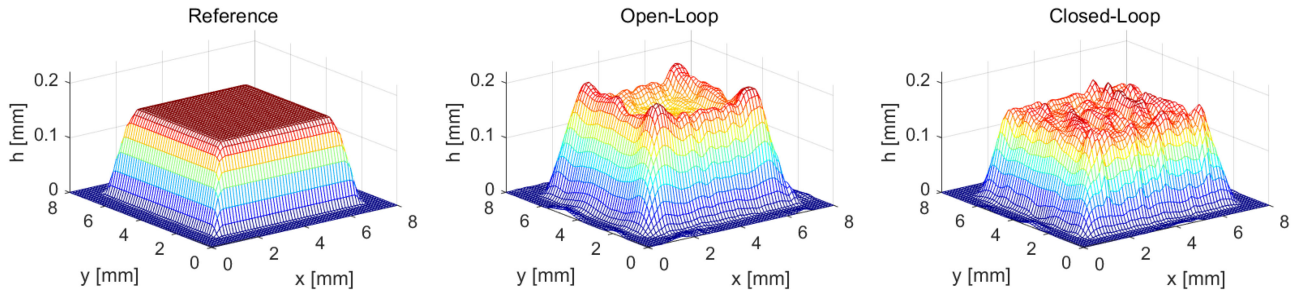


Fig. 6. Printing results comparison between open-loop printing and distributed MPC printing for the square prism.

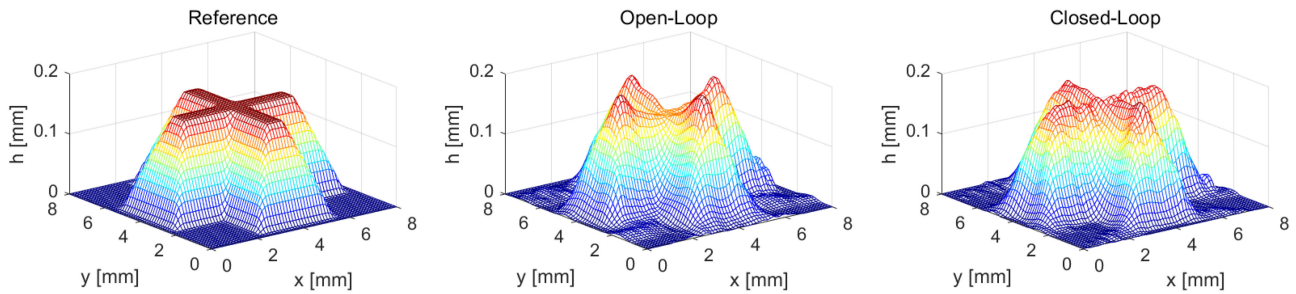


Fig. 7. Printing results comparison between open-loop printing and distributed MPC printing for the cross-shaped part.

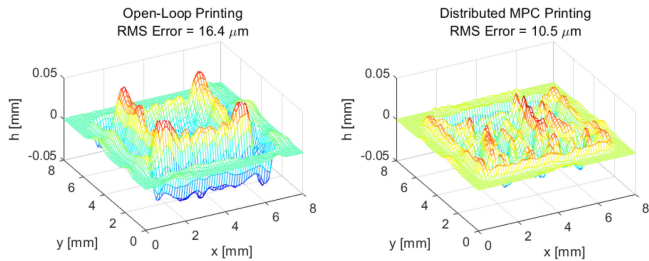


Fig. 8. Error profiles of open-loop printing and distributed MPC printing for the square prism.

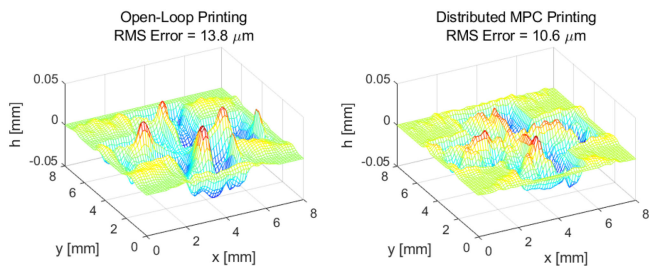


Fig. 9. Error profiles of open-loop printing and distributed MPC printing for the cross frustum.

obtained with open-loop printing, while the corresponding RMS errors using distributed MPC are $10.5 \mu\text{m}$, $6.9 \mu\text{m}$, and $10.6 \mu\text{m}$. Similar performance improvement is noted for side-wall error accuracy as well.

TABLE V
COMPARISON OF RMS ERROR OF PRINTED PARTS

Part Shape	Overall RMS Error (μm)		%	RMS Error - Side Wall (μm)	
	OL	CL		OL	CL
Square prism	16.4	10.5	36	12.1	8.7
Square frustum	11.3	6.9	39	9.1	5.2
Cross frustum	13.8	10.6	23	13.1	10.1

The first row and third rows displays results from parts shown earlier.

C. Layer-to-Layer Behavior

An interesting phenomenon observed during open-loop printing (see Figs. 6 and 7, middle plot) is that the height profile typically exhibits a bulging effect at the edges. For the open-loop printed cross in Fig. 7, each arm, in fact slopes upward though the droplets are evenly distributed. This behavior is due to the fluid's surface tension and is the main source of the deviation of the part from the reference. Fig. 10 shows cross-sections through the first, third, and fifth layers of the square and cross frustums for both the open-loop and closed-loop cases. For the open-loop printing case, the cross sections indicate that the bulge due to surface tension is growing across the layers (see Fig. 10 top). However, the evolution of the part never becomes unstable since volume is conserved. On the other hand, the distributed MPC compensates for the surface tension dynamics allowing the part conform to the reference geometry through all the layers (see Fig. 10 middle). This is highlighted in the layer to layer RMS error evolution plot (see Fig. 10 bottom) comparing open-loop with closed-loop control.

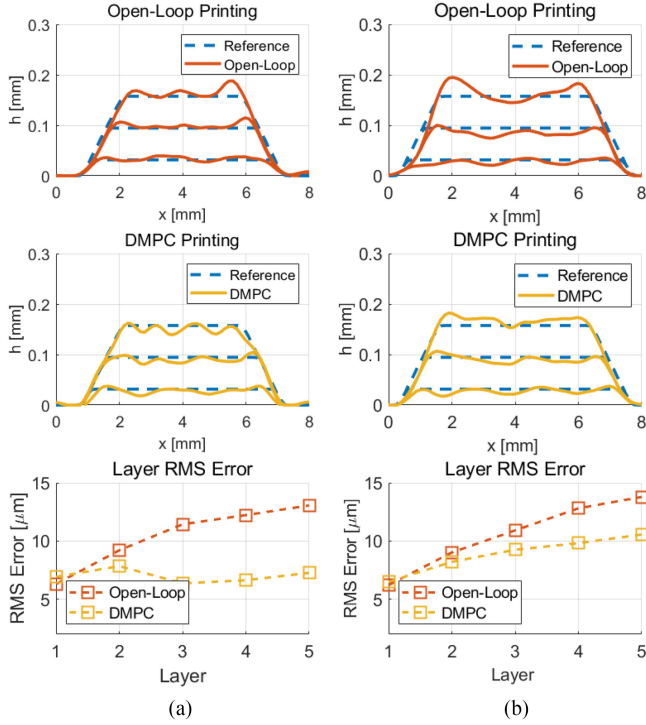


Fig. 10. Comparison between cross-sections of open-loop printing and distributed MPC printing for square and cross frustums. Cross-sections through the first, third, and fifth layers are shown. The RMS Error across the five layers is plotted at the bottom.

D. Discussion

By implementing the distributed MPC algorithm for inkjet 3-D printing, we have demonstrated the following:

- 1) For practical printing applications with large grid sizes, predictive feedback control of the process is numerically tractable and may be efficiently solved on a standard personal computer.
- 2) The distributed controller is demonstrably capable of handling disturbances and model uncertainties associated with a variety of actual printing scenarios.
- 3) The time required for solving the DMPC to determine the control input is much smaller than print time. Typical print time for a 64×64 grid size layer is about 3 min while the control algorithm implementation (including feedback measurement layer scanning and computation) requires about 15 s.

The value of implementing feedback control in 3-D printing is dependent on application; that is, the additional cost and time required for feedback should be offset in value by the precision added by control. In this article, we have demonstrated that we can in fact substantially improve the printing precision in inkjet 3-D printing on the geometry level using MPC, and that the otherwise computational expense associated with such geometry-level feedback control can be sufficiently expedited so this improvement in printing precision is practicable in real time. This result is particularly relevant in 3-D printing of polymers (as conducted in this work) since polymer parts, unlike printed metal parts, may not require or be suitable for mechanical finishing.

VIII. CONCLUSION

This article investigated advanced modeling and closed-loop control for high resolution inkjet 3-D printing. Specifically, this article has presented the printing control objective as a graph-based MPC problem. Then, a distributed control scheme is used to efficiently solve this computationally expensive problem. We demonstrated that this distributed control scheme offers a similar performance as centralized MPC at significantly reduced expense. The scheme was experimentally validated in closed-loop printing of various parts and showed substantial improvement in printing accuracy over open-loop printing.

APPENDIX

We start with the assumption there exists an optimal control input u_L^* such that $r_{L+1} = \mathcal{A}r_L + \mathcal{B}u_L^*$. Let $x_L = h_L - r_L$ and $w_L = u_L - u_L^*$, then the cost function in (22) is

$$J(x_L, W_L) = x_{N|L}^T P x_{N|L} + \sum_{i=0}^{N-1} [x_{i|L}^T Q x_{i|L} + w_{i|L}^T R w_{i|L}] \quad (26)$$

where $x_{0|L} = x_L$. Thus, we want to show that the closed-loop MPC system $x_{L+1} = \mathcal{A}x_L + \mathcal{B}K_{\text{MPC}}(x_L)$ is asymptotically stable at the equilibrium origin. Here, the following Lyapunov function is used to prove the stability

$$V(x_L) = \min_{W_L} J(x_L, W_L). \quad (27)$$

Since $Q \succ 0$ and $R \succ 0$, it follows that:

- 1) $V(0) = 0$;
- 2) $V(\xi) \leq \xi^T Q \xi, \forall \xi \neq 0$;
- 3) $V(\xi) \rightarrow \infty$ as $\|\xi\| \rightarrow \infty$.

We need to show that

$$V(x_{L+1}) - V(x_L) < 0 \quad \forall x_L \neq 0. \quad (28)$$

Consider the optimal input sequence

$$W_L^*(x_L) = \arg \min_{W_L} J(x_L, W_L) = \{w_{0|L}^*, w_{1|L}^*, \dots, w_{N-1|L}^*\}. \quad (29)$$

The following shifted input sequence at layer $L+1$ is

$$\tilde{W}_{L+1}(x_{L+1}) = \{w_{1|L}^*, w_{2|L}^*, \dots, w_{N-1|L}^*, Kx_{N|L}^*\} \quad (30)$$

Note $\tilde{W}_{L+1}$ is not necessarily the *optimal* input at layer $L+1$ for x_{L+1} . Let $\tilde{V}(x_{L+1}) = J(x_{L+1}, \tilde{W}_{L+1})$, we have

$$\begin{aligned} \tilde{V}(x_{L+1}) - V(x_L) &= -l(x_L, w_{0|L}^*) - f_T(x_{N|L}^*) \\ &\quad + l(x_{N|L}^*, Kx_{N|L}^*) + f_T((A+BK)x_{N|L}^*) \end{aligned} \quad (31)$$

where $l(a, b) = a^T Q a + b^T R b$ is the stage cost, $f_T(a) = a^T P a$ is the terminal cost. If we can show the sum of the last three terms is nonpositive, then

$$\tilde{V}(x_{L+1}) - V(x_L) \leq -l(x_L, w_{0|L}^*) \leq -x_L^T Q x_L. \quad (32)$$

Let ξ represent $x_{N|L}^*$, then the sum of the last three terms is

$$\xi^T ((A+BK)^T P (A+BK) - P + Q + K^T R K) \xi. \quad (33)$$

Since we have

$$(A + BK)^T P(A + BK) - P \preceq -Q - K^T R K \quad (34)$$

as a condition in (25), the sum of last three terms is nonpositive, hence

$$\tilde{V}(x_{L+1}) - V(x_L) \leq -x_L^T Q x_L < 0 \quad \forall x_L \neq 0. \quad (35)$$

Recall that $\tilde{W}_{L+1}$ is not necessarily the optimal at layer $L + 1$ for x_{L+1} , thus

$$V(x_{L+1}) - V(x_L) \leq \tilde{V}(x_{L+1}) - V(x_L) < 0 \quad \forall x_L \neq 0 \quad (36)$$

which completes the proof. ■

REFERENCES

- [1] P. M. Sammons, M. L. Gegel, D. A. Bristow, and R. G. Landers, "Repetitive process control of additive manufacturing with application to laser metal deposition," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 566–575, Mar. 2019.
- [2] L. Tang and R. G. Landers, "Melt pool temperature control for laser metal deposition Processes—Part I: Online temperature control," *J. Manuf. Sci. Eng.*, vol. 132, no. 1, pp. 1087–1357, Jan. 2010.
- [3] A. Herali, A.-K. Christiansson, and B. Lennartson, "Height control of laser metal-wire deposition based on iterative learning control and 3D scanning," *Opt. Lasers Eng.*, vol. 50, no. 9, pp. 1230–1241, 2012.
- [4] A. Fathi, A. Khajepour, M. Durali, and E. Toyserkani, "Geometry control of the deposited layer in a nonplanar laser cladding process using a variable structure controller," *J. Manuf. Sci. Eng.*, vol. 130, no. 3, pp. 1087–1357, May 2008.
- [5] X. Zhao and D. W. Rosen, "An implementation of real-time feedback control of cured part height in exposure controlled projection lithography with in-situ interferometric measurement feedback," *Additive Manuf.*, vol. 23, pp. 253–263, 2018.
- [6] P. Sitthi-Amorn *et al.*, "MultiFab: A machine vision assisted platform for multi-material 3D printing," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 129.
- [7] K. Barton, S. Mishra, A. Alleyne, P. Ferreira, and J. Rogers, "Control of high-resolution electrohydrodynamic jet printing," *Control Eng. Pract.*, vol. 19, no. 11, pp. 1266–1273, 2011.
- [8] J. Mazumder, A. Schifferer, and J. Choi, "Direct materials deposition: Designed macro and microstructure," *Material Res. Innov.*, vol. 3, no. 3, pp. 118–131, 1999.
- [9] E. Skordeli and C. Doumanidis, "Infinite-dimensional geometric regulation in solid freeform fabrication," in *Proc. IEEE Amer. Control Conf.*, 1999, vol. 3, pp. 1734–1738.
- [10] D. L. Cohen and H. Lipson, "Geometric feedback control of discrete-deposition SFF systems," *Rapid Prototyping J.*, vol. 16, no. 5, pp. 377–393, 2010.
- [11] 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.
- [12] Y. Guo and S. Mishra, "A predictive control algorithm for layer-to-layer ink-jet 3D printing," in *Proc. Amer. Control Conf.*, 2016, pp. 833–838.
- [13] 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, 2018.
- [14] C. Pannier, M. Wu, D. Hoelzle, and K. Barton, "LPV models for jet-printed heightmap control," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 5402–5407.
- [15] L. Lu, J. Zheng, and S. Mishra, "A layer-to-layer model and feedback control of ink-jet 3-D printing," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 3, pp. 1056–1068, Jun. 2015.
- [16] Y. Guo, J. Peters, T. Oomen, and S. Mishra, "Distributed model predictive control for ink-jet 3D printing," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatron.*, 2017, pp. 436–441.
- [17] Y. Guo, J. Peters, T. Oomen, and S. Mishra, "Control-oriented models for ink-jet 3D printing," *Mechatronics*, vol. 56, pp. 211–219, May 2018.
- [18] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill Publishing, 2012.
- [19] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.
- [20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [21] P. Giselsson and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," in *Proc. Conf. Decis. Control*, 2010, pp. 7272–7277.
- [22] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.
- [23] C. Chen and L. Shaw, "On receding horizon feedback control," *Automatica*, vol. 18, no. 3, pp. 349–352, 1982.
- [24] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [25] S. A. Keerthi and E. G. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *J. Optim. Theory Appl.*, vol. 57, no. 2, pp. 265–293, 1988.
- [26] J. B. Rawlings and K. R. Muske, "The stability of constrained receding horizon control," *IEEE Trans. Autom. Control*, vol. 38, no. 10, pp. 1512–1516, Oct. 1993.



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

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



Yijie Guo received the B.S. degree from the Zhejiang University, Hangzhou, China, in 2013, and the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2017, both in mechanical engineering.

He is currently a Senior Algorithm Engineer with UBTECH Robotics, Beijing, China. His research interests include the areas of classical and modern control, optimization, and machine learning.



Joost Peters received the M.Sc. degree in mechanical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2018.

He is currently a Scientist Innovator with the Netherlands Organization for Applied Scientific Research (TNO), focusing on dynamics and control for laser satellite communication systems, atomic force microscopes and optomechatronic systems. His research interests include learning control, distributed optimization, model predictive control, and control for laser satellite communication systems.



Tom Oomen received the M.Sc. degree (*cum laude*) and the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2005 and 2010, respectively, both in mechanical engineering.

He held visiting positions with KTH, Stockholm, Sweden, and with The University of Newcastle, Callaghan NSW, Australia. He is currently an Associate Professor with the Department of Mechanical Engineering, Eindhoven University of Technology. His research interests include the field of data-driven modeling, learning, and control, with applications in precision mechatronics.



Sandipan Mishra received the B.Tech. degree from the Indian Institute of Technology Madras, Chennai, India, in 2002, and the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2008, both in mechanical engineering.

He is currently an Associate Professor with the Department of Mechanical, Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA. His current research interests include general area of control theory, iterative learning control, optimal control, and precision mechatronics, as applied to autonomous aerial vehicles, additive manufacturing, and smart building systems.