2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)
Sheraton Arabella Park Hotel, Munich, Germany, July 3-7, 2017

Distributed Model Predictive Control for Ink-Jet 3D Printing

Yijie Guo[1], Joost Peters[2], Tom Oomen[2] and Sandipan Mishra[1]

Abstract— **This paper develops a closed-loop approach for ink-jet 3D printing. The control design is based on a distributed model predictive control scheme, which can handle constraints (such as droplet volume) as well as the large-scale nature of the problem. The high resolution of ink-jet 3D printing make centralized methods extremely time-consuming, thus a distributed implementation of the controller is developed. First a graph-based height evolution model that can capture the liquid flow dynamics is proposed. Then, a scalable closed-loop control algorithm is designed based on the model using Distributed MPC, that reduces computation time significantly. The performance and efficiency of the algorithm are shown to outperform open-loop printing and closed-loop printing with existing Centralized MPC methods through simulation results.**

I. INTRODUCTION

Additive Manufacturing (AM) is a class of manufacturing processes in which material is added layer-upon-layer to construct 3D objects. Recently, AM has seen a significant increase in popularity both in commercial applications and research [1]. AM processes consist of many different technologies. In this paper, ink-jet 3D printing is considered, which has been widely applied in commercial printers for its simplicity and high resolution.

Ink-jet 3D printers build 3D objects by jetting photopolymer layer-upon-layer with UV (ultra-violet) light curing in between. Currently, this process is typically performed in an open-loop manner, in which the number of layers to be deposited and the droplet patterns for each layer are determined in advance. This open-loop approach is vulnerable to uncertainties in droplet sizes, shapes, and locations since it does not use any feedback of the height profile during printing. This can result in undesired part geometry where the printed shape poorly matches the desired geometry [2]. Meanwhile, high resolution of ink-jet 3D printing (x-y resolution is $0.125\mathrm{mm}$, for the particular printer used in this study) makes controlling this process a large scale problem. To account for uncertainties in the printing process and the large scale of the control problem, a closed-loop control algorithm that is scalable should be proposed.

Standard ink-jet 2D printing has been well studied in terms of voltage waveform design for generating consistent droplets [3], [4] and medium deformation compensation [5]. But for ink-jet 3D printing where height is concerned, these techniques are insufficient. Various approaches have been proposed for the control of ink-jet 3D printing or similar processes. A spatial iterative learning control algorithm is proposed and implemented in [6], [7] for electrohydrodynamic jet printing. In [8], iterative learning control and a feedback controller are combined to regulate the jetting frequency, thus to improve printing accuracy. In [9] a *Greedy Geometric Feedback* algorithm that iteratively searches for locations to deposit droplets based on the geometry tracking error is proposed. Although this approach addresses the whole-part geometry, it does not account for droplets' influence on neighbors. Furthermore, the greedy search algorithm is typically poorly scalable. In [10], [2], a nonlinear empirical model that considers material flow and a predictive control algorithm are proposed. This control algorithm uses brute-force search to minimize a cost function, and scalability of the control problem is not considered. In [11], a simplified linear height evolution model is proposed based on the 2D model in [6]. A predictive control algorithm is designed that aims to solve a quadratic program each layer, which is more efficient than brute-force search. However, it still suffers from poor scalability of the control problem. Thus, the aim of this research is to address both uncertainties in the printing process and the large scale of the control problem by proposing a control-oriented layer height evolution model and designing a scalable closed-loop control algorithm.

The main contribution of this paper is the development of a control-oriented linear model that accounts for material flow during the printing process, and a scalable closed-loop control algorithm based on distributed model predictive control (MPC) techniques [12], that reduces the computation time for control significantly. We show that the combination of advanced modeling and Distributed MPC strategies is valuable, and potentially enables closed-loop high resolution ink-jet 3D printing. The paper is organized as follows. First, the general printing control problem is described in Sec. II. Next, the proposed model is presented in Sec.III. Then the Distributed MPC based control algorithm is proposed in Sec. IV. Finally, in Sec. V simulation results are presented to compare Distributed MPC with open-loop printing and closed-loop printing using Centralized MPC.

II. PROBLEM DESCRIPTION

The general formulation of the printing control problem is presented in this section. The printing region is discretized into a $n_x \times n_y$ size grid based on the printing resolution, the number of points in the grid is $n = n_x \cdot n_y$. Fig. 1 illustrates the closed-loop layer-to-layer printing process for a 3×3 grid example, in which H_d, H_L and $U_L \in R^{n_x \times n_y}$ are matrices.

[1]Yijie Guo and Sandipan Mishra are with the Department of Mechanical, Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA guoy7@rpi.edu and mishrs2@rpi.edu

[2]Joost Peters and Tom Oomen are with the Department of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands j.peters.2@student.tue.nl and T.A.E.Oomen@tue.nl

To do matrix multiplication, their vector forms are used in modeling and control. Thus, the final and current layer height profiles are denoted by $h_d \in \mathbb{R}^n$ and $h_L \in \mathbb{R}^n$, where L indicates the layer number. At each layer an input sequence (droplet pattern) $u_L \in [V_{min}, V_{max}]^n$ is applied, which indicates the droplet volume at each location, where V_{min} and V_{max} are the minimum and maximum droplet volume that can be controlled. The control objective is to generate a droplet pattern $u_L \in [V_{min}, V_{max}]^n$, that minimizes the geometric tracking error $\|e\| = \|h_L - r\|$, based on feedback of the height profile. In Sec. IV, this control problem is described in more mathematical details.

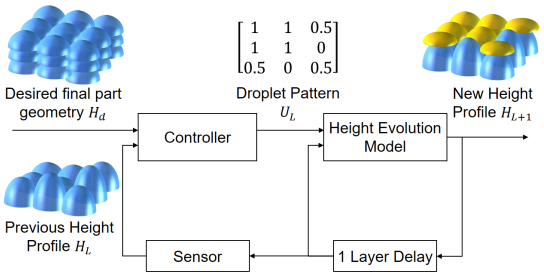

Fig. 1. A schematic block diagram of the closed-loop layer-to-layer printing process.

To accurately control the geometric shape of the printed part, a model that can capture the layer-to-layer height evolution is required. Thus, in the next section, a graph-based layer height evolution model is presented.

III. Model Description

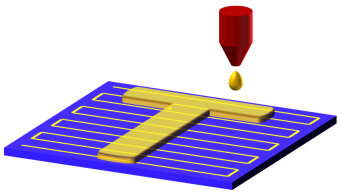

Fig. 2. Printing process for one layer. Nozzle moves along a predetermined path and deposits droplets sequentially.

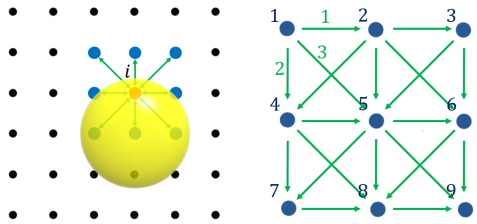

Fig. 3. The left shows node i's height change at time step k is caused by material flow between neighbors and material deposition at time step k. The right is an example of a directed graph for a 3×3 grid. Incidence matrix D describes the relationship between links and nodes. For example, $D(1,1) = -1$ because link 1 starts at node 1, according to Eq. (2).

In this section, the layer height evolution model in Fig. 1 is proposed, which is a graph-based model that captures the flow of the deposited material. This model is inspired by lumped heat transfer models, similar to those used in building temperature control [13]. The key idea is that material flows from higher to lower heights proportional to the height difference, like heat transfers from higher to lower temperatures proportional to the temperature difference.

In the printing process, droplets are sequentially deposited along a predetermined printing path, which usually is a raster path [14], as shown in Fig. 2. Each time step, the nozzle moves to a certain location according to the printing path. As shown in Fig. 3, the height change at node i at time step k is caused by material flow and material deposition, it is described by:

$$\Delta h_{i,k} = -\sum_{j \in \mathcal{N}_i} K_{ij}(h_{i,k} - h_{j,k}) + B_{i,k}u_k, \qquad (1)$$

where $\mathcal{N}_i$ is the set of neighbors of node i. The first term $-\sum_{j \in \mathcal{N}_i} K_{i,j}(h_{i,k} - h_{j,k})$ captures the effect that the deposited liquid material will flow from higher location to lower location, $K_{ij} > 0$ represents the flowability parameter that describes how much the liquid will flow based on the height difference with neighbouring locations. The second term $B_{i,k}u_k$ captures the material deposition, $B_{i,k}$ is the height increase at node i caused by unit size droplet at time step k as shown in Fig. 4, u_k is the droplet volume at time step k.

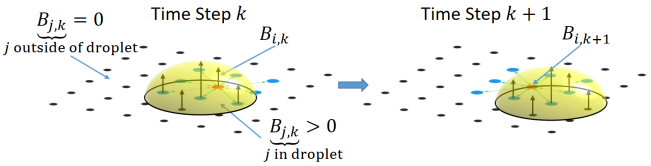

Fig. 4. $B_{i,k}$ is the height increase at node i caused by unit size droplet at time step k, all nodes' height increase caused by this unit size droplet constructs a $n \times 1$ vector B_k. If node j is in the droplet, $B_{j,k} > 0$, if node j is outside of the droplet, $B_{j,k} = 0$. Note that B_k is updated with time step, as the droplet location will change.

The incidence matrix D captures the nodes connectivity (through links) in a directed graph, as shown in Fig. 3 on the right. The elements of D are defined by

$$D(p,q) = \begin{cases} 1, & \text{if link } q \text{ ends at node } p \\ -1, & \text{if link } q \text{ starts at node } p \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

This allows us to combine (1), and (2) into (3), which is the whole grid height change model at time step k:

$$h_{k+1} = h_k - DK_kD^Th_k + B_ku_k, \qquad (3)$$

where $h_k \in \mathbb{R}^n$ is a column vector that defines the height profile of the layer when the nozzle is at the printing path time step k, $D \in \mathbb{R}^{n \times l}$, with l the number of links, is the incidence matrix. Diagonal positive definite matrix $K \in \mathbb{R}^{l \times l}$ contains the flowability parameters. Here, K_k is updated with time step k to set the links that are influenced by the flow dynamics. The height increase of the full grid caused by a

unit size droplet at time step k, constructs a $n \times 1$ vector B_k as shown in Fig. 4.

A. Layer Evolution Model

The model described above shows the height evolution from *time step* to *time step*. We now present a layer-to-layer model derived from this model through lifting. First, the time step height evolution model (3) is rewritten as:

$$h_{k+1} = A_k h_k + B_k u_k, \tag{4}$$

where $A_k = (I - DK_kD^T)$. At the final time step n of printing one layer, the height profile can be calculated as:

$$h_n = (\prod_{i=n}^{1} A_i)h_1 + \begin{bmatrix} (\prod_{i=n}^{2} A_i)B_1 \\ (\prod_{i=n}^{3} A_i)B_2 \\ \vdots \\ A_nB_{n-1} \\ B_n \end{bmatrix}^T \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}, \tag{5}$$

Assume we are printing the $(L+1)^{th}$ layer, the first time step height profile h_1 is actually the final height profile h_L of the L^{th} layer printing, the final time step height profile h_n is actually the final height profile h_{L+1} of the $(L+1)^{th}$ layer printing. From now on, h_L only represents the layer height profile, the time step height profile notation h_k is disused.

Then, we can have our layer-to-layer height evolution model (with L as the layer number):

$$h_{L+1} = \mathcal{A}h_L + \mathcal{B}u_L, \tag{6}$$

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

$$\begin{bmatrix}(\prod_{i=n}^{2} A_i)B_1 & (\prod_{i=n}^{3} A_i)B_2 & \cdots & A_nB_{n-1} & B_n\end{bmatrix},$$

the L^{th} layer control input u_L is

$$\begin{bmatrix}u_1 & u_2 & \cdots & u_{n-1} & u_n\end{bmatrix}^T,$$

the k^{th} element is the k^{th} time step's control input when printing this layer. If the nozzle moves along a certain path, then $\mathcal{A}$, $\mathcal{B}$ are fixed for each layer.

Thus, we now have a layer-to-layer model that accounts for material flow during the printing process. With this lifted description of the linear time-invariant layer-to-layer height evolution model, an MPC design strategy can be developed.

IV. Distributed Model Predictive Control

In this section, the controller in Fig. 1 is designed. A Distributed MPC algorithm is developed that provides the ability to solve the large control problem efficiently. First, the standard Centralized MPC problem is described. Then, this centralized control problem is partitioned according to the partitioning of the printing region. Finally, the distributed algorithm using dual decomposition is presented. In Sec. V it will be shown that this method reduces the computation time for layer-to-layer control significantly.

A. Centralized MPC

The control problem introduced in Sec. II is cast into an MPC framework that solves an optimization problem over a finite receding horizon of N *layers* each layer. The optimization problem can be defined as:

$$\begin{aligned} &\min_{U_L} J(h_L, U_L) \\ &\text{s.t.} \quad \mathcal{E}U_L \leq c, \end{aligned} \tag{7}$$

where $U_L = [u_{0|L}^T \ ... \ u_{N-1|L}^T]^T$, $u_{i|L}$ indicates the i^{th} layer control input in the receding horizon. The cost function is designed to penalize tracking error:

$$\begin{aligned} J(h_L, U_L) = &(h_{N|L} - r_{N|L})^T P (h_{N|L} - r_{N|L}) \\ &+ \sum_{i=0}^{N-1} \left[(h_{i|L} - r_{i|L})^T Q (h_{i|L} - r_{i|L})\right], \end{aligned} \tag{8}$$

where $h_{i|L}$ indicates the i^{th} layer height profile in the receding horizon. P and Q are (semi)positive-definite matrices, Q is the state cost matrix and P is the terminal cost matrix that can be designed to guarantee MPC stability. In (7), $\mathcal{E}$ and c are a matrix and column vector that are defined by:

$$\mathcal{E} = \begin{bmatrix} E_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & E_{N-1} \end{bmatrix} \quad c = \begin{bmatrix} b_0 \\ \vdots \\ b_{N-1} \end{bmatrix}, \tag{9}$$

where $E_i = \begin{bmatrix}-I & I\end{bmatrix}^T$ and $b_i = \begin{bmatrix}-u_{\text{low}} & u_{\text{high}}\end{bmatrix}^T$ that defines the upper and lower constraints on the input. The optimization is performed each layer, and $u^\star_{0|L}$ is applied.

Centralized MPC methods with quadratic convex problems are well developed [15][16]. However, for increasing size of the optimization problem (prediction horizon N, grid size n) computation time of standard MPC can become tremendous. Dense (Centralized) formulations (where the cost function is described solely as function of the (future) inputs), scale with $O(N^3n^3)$ using interior-point convex solvers [17]. This paper proposes to use a distributed approach to the MPC control problem to reduce computation time.

B. Partitioning of the optimization problem

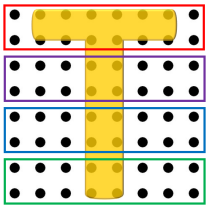

Fig. 5. To decompose the optimization problem, the whole printing region is separated into l sub-regions, here is an example of 4 sub-regions.

First, the whole printing region is separated into l sub-regions, as shown in Fig. 5. The layer height evolution model (6) introduced in section III is then partitioned based on the region separation:

$$\begin{bmatrix} h_{L+1}^1 \\ \vdots \\ h_{L+1}^l \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} & \cdots & \mathcal{A}_{1l} \\ \vdots & \ddots & \vdots \\ \mathcal{A}_{l1} & \cdots & \mathcal{A}_{ll} \end{bmatrix} \begin{bmatrix} h_L^1 \\ \vdots \\ h_L^l \end{bmatrix} + \begin{bmatrix} \mathcal{B}_{11} & \cdots & \mathcal{B}_{1l} \\ \vdots & \ddots & \vdots \\ \mathcal{B}_{l1} & \cdots & \mathcal{B}_{ll} \end{bmatrix} \begin{bmatrix} u_L^1 \\ \vdots \\ u_L^l \end{bmatrix}. \tag{10}$$

For the partitioned system (10), the cost function becomes

$$J(h_L, U_L) = \sum_{j=1}^{l} \Big(\sum_{i=0}^{N-1} \Big[(h^j_{i|L} - r^j_{i|L})^T Q_j (h^j_{i|L} - r^j_{i|L}) \Big] + (h^j_{N|L} - r^j_{N|L})^T P_j (h^j_{N|L} - r^j_{N|L}) \Big). \tag{11}$$

A partitioned quadratic program is constructed for the MPC problem:

$$\begin{aligned} \min_x \quad & \sum_{j=1}^{l} x_j^T H_j x_j \\ \text{subject to} \quad & \sum_{p=1}^{l} F_{jp} x_p = z_j, \ j \in \{1, \cdots, l\} \\ & T_j x_j \leq q_j, \ j \in \{1, \cdots, l\}, \end{aligned} \tag{12}$$

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

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \end{bmatrix}, \ x_j = \begin{bmatrix} h^j_{0|L} - r^j_{0|L} \\ \vdots \\ h^j_{N|L} - r^j_{N|L} \\ \hline u^j_{0|L} \\ \vdots \\ u^j_{N-1|L} \end{bmatrix} \ \forall j \in \{1, \cdots, l\}. \tag{13}$$

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

$$H_j = \left[\begin{array}{cccc|c} Q_j & 0 & 0 & 0 & \underline{0} \\ 0 & \ddots & 0 & 0 & \vdots \\ 0 & 0 & Q_j & 0 & \vdots \\ 0 & 0 & 0 & P_j & \vdots \\ \hline \underline{0} & \cdots & \cdots & \cdots & \underline{0} \end{array} \right] \forall j \in \{1, \cdots, l\}. \tag{14}$$

The equality constraint $\sum_{p=1}^{l} F_{jp} x_p = z_j$ for $j \in \{1, \cdots, l\}$ is the separated form of $Fx = z$, which captures the layer height evolution dynamics, F_{jp} and z_j can be constructed by reformulation of (10) with the new variable x in (13).
The inequality constraint $T_j x_j \leq q_j$ captures the upper and lower bounds on the input (9), T_j and q_j are described by:

$$T_j = \left[\ \underline{0} \ \middle| \ \mathcal{E}_j \ \right], \ q_j = c_j, \ \forall j \in \{1, \cdots, l\}. \tag{15}$$

This formulation remains 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 into subproblems that can be solved in parallel. This is the key feature of dual decomposition that allows for distributed optimization. The Distributed MPC algorithm using dual decomposition is presented in the next section.

C. Distributed MPC

This section provides the algorithm for Distributed MPC for the height tracking problem. The algorithm uses dual decomposition to solve the decomposable optimization problem of section IV B. The Lagrangian dual problem of the separable optimization problem (12) is given by [12]:

$$\begin{aligned} \max_{\lambda} \quad & \min_x \sum_{j=1}^{l} \left[x_j^T H_j x_j + \lambda_j^T \left(\sum_{p=1}^{l} [F_{jp} x_p] - z_j \right) \right] \\ & \text{subject to} \quad T_j x_j \leq q_j, \ j \in \{1, \cdots, l\} \end{aligned} \tag{16}$$

This optimization problem can be rewritten to:

$$\max_{\lambda} \quad \sum_{j=1}^{l} \underbrace{\left[\min_{x_j} \quad x_j^T H_j x_j + x_j^T \sum_{p=1}^{l} [F_{pj} \lambda_p] - \lambda_j^T z_j \right]}_{\text{subject to} \quad T_j x_j \leq q_j \qquad L_j} \tag{17}$$

It is observed that the resulting minimization of L_j is again a Quadratic Program, which can be solved for each subproblem. When the maximum in (17) is obtained, the constraints are satisfied by definition of Lagrangian duality. It is observed that the minimization step is completely decentralized, for given 'prices' λ. However, finding these optimal prices requires coordination which is done by a 'price update' through gradient ascent [12]. The price update during the iterations of dual decomposition is described by:

$$\lambda^{s+1} = \lambda^s + \gamma^s \nabla g^s(\lambda), \tag{18}$$

where $\nabla g^s(\lambda) = F x^s - z$, s is iteration number. To converge to the optimal price $\lambda^\star$, it is necessary to determine the appropriate step size sequence γ^s. One traditional approach proposed for gradient descent is the approach introduced in 1988 by Barzilai and Borwein [18]. For gradient ascent, this method computates the gradient step size as:

$$\gamma^s = \frac{-(\nabla g^s - \nabla g^{s-1})^T (\lambda^s - \lambda^{s-1})}{(\nabla g^s - \nabla g^{s-1})^T (\nabla g^s - \nabla g^{s-1})} \tag{19}$$

which is an approximation of the Newtons method, where the inverse of the Hessian is used [18]. By approximating, it avoids the expensive computation of the Hessian, but still achieves good convergence. In [19] it is observed that the rate of convergence is R-superlinear with an order of $\sqrt{2} - \epsilon$ with $\epsilon > 0$ any small number.
With the defined local minimization in (17) and price update method in (18) and (19), an algorithm for Distributed MPC can be constructed. The algorithm is summarized by the pseudo code presented in Algorithm 1. In this algorithm the following steps are recognized:

- 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 $\lambda^{S_k}_{L-1}$, the MPC problem matrices (H, F, T, q), the number of partitions l and the price convergence criterion Λ.
- In lines 5-7 the local minimization is performed based on the current price λ^s.
- In lines 8-10 the price update is performed by a gradient ascent step. To this end the Barzilai-Borwein method is

used to determine the step size.

- In lines 11-16 the stopping condition is posed, based on convergence of the price λ. If the algorithm is converged, 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}, Λ_λ
2: **Output:** u_L,
3: **Initialize:** $\gamma^{(0)}$, $\lambda_L^{(0)} = \lambda_{L-1}^{S_k}$
4: **for** $s = 1, 2, \cdots, s_{max}$ **do**
5: **for** $j = 1, 2, \cdots, l$ **do**
6: **Solve** $x_j^{s+1} =$

$$\text{argmin}_{x_j} x_j^T H_j x_j + x_j^T \sum_{p=1}^{l} \left[F_{pj} \lambda_p^s\right] - (\lambda_j{}^s)^T z_j$$

$$\text{subject to} \quad T_j x_j \leq q_j$$

7: **end for**
8: **Compute** $\nabla g^s(\lambda) = F x^s - z$
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** $\lambda^{s+1} = \lambda^s + \gamma^s \nabla g^s(\lambda)$
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**

V. Simulation Results

This section presents simulation results of open-loop (OL), Centralized MPC (CMPC) and Distributed MPC (DMPC) printing. These examples indicate the benefit of closing the loop, in presence of uncertainty. Results are presented that show the comparison of performance, in terms of layer height tracking and computation time for OL, CMPC and DMPC printing. In these comparisons, different grid sizes and prediction horizons are considered.

It is observed that during the printing process the droplet shape and volume vary based on operating environmental conditions, which introduces uncertainty to the droplet model. To capture this uncertainty, we constructed a droplet samples set that contains 167 droplet shapes, measured by a LJ-G030 2D height sensor on an experimental setup. In the simulation of the printing process, for each droplet deposition, first a droplet sample is randomly selected from the samples set, then the corresponding B_k is constructed based on this droplet shape. In the prediction of the control algorithm, the average shape of the samples set is used as the droplet shape.

First, open-loop is compared with CMPC and DMPC in terms of layer height tracking. Figure 6 shows results for open-loop printing and closed-loop MPC printing. For this example, the grid size is 60×60, and the prediction horizon of the MPC algorithm $N = 5$. The desired layer height $h_d = L \cdot 0.03mm$, where $L = 5$ is the layer number.

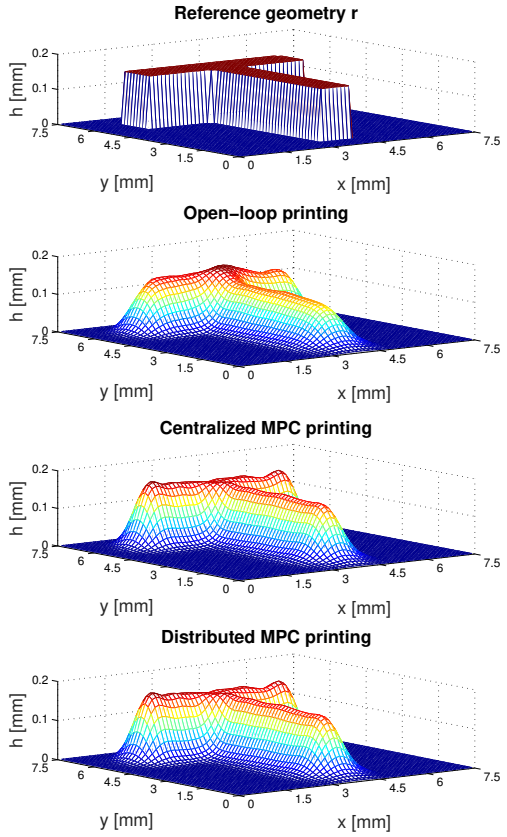

Fig. 6. Comparison of open-loop and closed-loop MPC printing of a 5 layer T-shaped geometry. It is observed that tracking of the height profile is improved for closed-loop printing. Furthermore it can be concluded that the Distributed MPC performs similar to Centralized MPC.

Table I summarizes the results in terms of layer height tracking for different grid sizes and prediction horizons. In bold the example of Figure 6 is presented.

TABLE I
2-norm of the layer height tracking $||e||_2$ for open-loop, Centralized MPC and Distributed MPC

	OL	CMPC			DMPC		
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)
40	1.23	1.09	1.05	1.04	1.07	1.05	1.05
50	1.40	1.25	1.20	1.19	1.25	1.20	1.19
60	**1.57**	1.43	1.40	**1.39**	1.43	1.39	**1.39**
80	1.95	1.80	1.75	1.75	1.79	1.75	1.75
100	2.38	2.18	2.12	2.13	2.18	2.11	2.13

It is observed that CMPC and DMPC improve the layer height tracking compared to open-loop printing. From Table I it is concluded that the DMPC controller provides similar reference tracking as the CMPC controller, since deviations in terms of layer height tracking error are small for all simulations conducted. This implies that the solution to the distributed problem is close to the centralized solution. The

drawback of the CMPC method is as mentioned before: the computational burden. Therefore, the computation time for the Centralized and Distributed MPC algorithms are compared for different grid sizes and prediction horizons. Figure 7 shows the comparison of the computation time for the two methods.

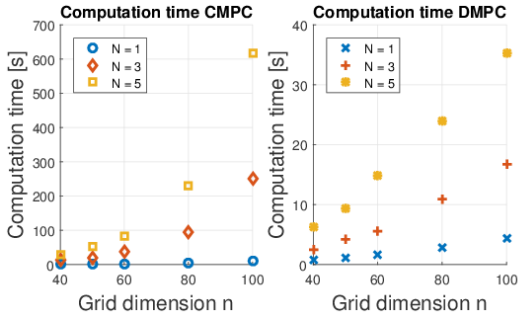

Fig. 7. Computation time comparison for Centralized MPC and Distributed MPC. It can be observed that for Distributed MPC, the computation time is significantly reduced.

The results of the computation time comparison are summarized in Table II. For comparison, the typical time of printing a 100×100 grid size square for one layer is 5 minutes. It is observed that a significant reduction in computation time is achieved by introducing the distributed algorithm. The distributed approach achieves lower computation times and better scaling in terms of grid size; it is observed that computation time only scales linear with the grid size in case of DMPC, which indicates the desired scalability.

TABLE II

AVERAGE COMPUTATION TIME FOR ONE LAYER FOR CENTRALIZED MPC AND DISTRIBUTED MPC.

	CMPC in [s]			DMPC in [s]		
n	N=1	N=3	N=5	N=1	N=3	N=5
40	0.43	12.89	29.13	0.72	2.48	6.20
50	0.73	20.68	51.88	1.15	4.14	9.36
60	1.20	36.68	**82.16**	1.61	5.58	**14.81**
80	3.89	95.24	230.31	2.91	10.84	24.02
100	11.58	249.63	616.57	4.30	16.78	35.29

VI. CONCLUSION

The results in this paper provide methods for closed-loop control of ink-jet 3D printing, enabling high-resolution 3D printing of complex geometries. More specifically, in this paper, a closed-loop control algorithm using Distributed MPC is proposed based on a graph-based layer height evolution model. The performance and efficiency of the algorithm is compared with open-loop printing and closed-loop printing using Centralized MPC, through simulation results. It is shown that the proposed algorithm has similar tracking performance as Centralized MPC, and achieves better performance than open-loop printing by addressing uncertainties in the printing process through feedback. In terms of the efficiency, the proposed algorithm reduces the calculation time significantly compared to Centralized MPC. The computation time only scales linearly with grid size, indicating the good scalability of the algorithm. Future work will be the experimental validation on a 3D printer setup.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation Career Award grant CMMI-1254313 and in part by the Center for Automation Technologies and Systems (CATS) under a block grant from the New York State Empire State Development Division of Science, Technology and Innovation (NYSTAR).

REFERENCES

[1] S. L. Ford, "Additive manufacturing technology: Potential implications for us manufacturing competitiveness," *Journal of International Commerce and Economics*, 2014.

[2] L. Lu, J. Zheng, and S. Mishra, "A layer-to-layer model and feedback control of ink-jet 3-d printing," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, 2015.

[3] M. G. Wassink, N. Bosch, O. Bosgra, and S. Koekebakker, "Enabling higher jet frequencies for an inkjet printhead using iterative learning control," in *Control Applications (CCA), 2005 IEEE Conference on*. IEEE, 2005, pp. 791–796.

[4] K. S. Kwon and W. Kim, "A waveform design method for high-speed inkjet printing based on self-sensing measurement," *Sensors and Actuators A: Physical*, vol. 140, no. 1, pp. 75–83, 2007.

[5] J. Bolder, T. Oomen, S. Koekebakker, and M. Steinbuch, "Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer," *Mechatronics*, vol. 24, no. 8, pp. 944–953, 2014.

[6] D. J. Hoelzle and K. L. Barton, "A new spatial iterative learning control approach for improved micro-additive manufacturing," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 1805–1810.

[7] Z. Wang, C. Pannier, L. Ojeda, K. Barton, and D. J. Hoelzle, "An application of spatial iterative learning control to micro-additive manufacturing," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 354–359.

[8] K. Barton, S. Mishra, A. Alleyne, P. Ferreira, and J. Rogers, "Control of high-resolution electrohydrodynamic jet printing," *Control Engineering Practice*, vol. 19, no. 11, pp. 1266–1273, 2011.

[9] D. L. Cohen and H. Lipson, "Geometric feedback control of discrete-deposition sff systems," *Rapid Prototyping Journal*, vol. 16, no. 5, pp. 377–393, 2010.

[10] L. Lu, J. Zheng, and S. Mishra, "A model-based layer-to-layer control algorithm for ink-jet 3d printing," in *Dynamic Systems and Control Conference (DSCC), 2014*. ASME, 2014, pp. V002T35A001–V002T35A001.

[11] Y. Guo and S. Mishra, "A predictive control algorithm for layer-to-layer ink-jet 3d printing," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 833–838.

[12] P. Giselsson and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," in *Conference on Decision and Control (CDC), 2010*. IEEE, 2010, pp. 7272–7277.

[13] H. Boyer, J. P. Chabriat, B. Grondin-Perez, C. Tourrand, and J. Brau, "Thermal building simulation and computer generation of nodal models," *Building and environment*, vol. 31, no. 3, pp. 207–214, 1996.

[14] S. C. Danforth and A. Safari, "Tool path-based deposition planning in fused deposition processes," *Journal of Manufacturing Science and Engineering*, vol. 124, no. 2, pp. 462–472, 2002.

[15] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Publishing, 2012.

[16] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Education, 2002.

[17] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 1, pp. 267–278, 2010.

[18] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[19] L. Qi, K. L. Teo, and X. Yang, *Optimization and control with applications*. Springer Science & Business Media, 2006, vol. 96.