Proceedings of the ASME 2023 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2023
August 20-August 23, 2023, Boston, USA

IDETC/CIE 2023-114959

PRINT AS A DANCE DUET: COMMUNICATION STRATEGIES FOR COLLISION-FREE ARM-ARM COORDINATION IN COOPERATIVE 3D PRINTING

Ronnie F. P. Stone
Walker Department of Mechanical Engineering
University of Texas at Austin
Austin, TX

Wenchao Zhou
Department of Mechanical Engineering
University of Arkansas
Fayetteville, AR

Ergun Akleman
J. Mike Walker '66 Department of Mechanical Engineering
Texas A&M University
College Station, TX

Vinayak R. Krishnamurthy
J. Mike Walker '66 Department of Mechanical Engineering
Texas A&M University
College Station, TX

Zhenghui Sha *
Walker Department of Mechanical Engineering
University of Texas at Austin
Austin, TX

ABSTRACT

One of the major challenges in 3D printing is its lack of scalability both in size and speed, which directly impacts its economic feasibility for large-scale industrial applications. Cooperative 3D printing (C3DP) is an emerging paradigm that aims to address these issues by employing multiple mobile printers that work in parallel. However, a crucial step in enabling C3DP is the development of a collision-free communication framework between the printers during the manufacturing process. Many C3DP systems found in the literature develop solutions for collision-free printing that are specific to the setup being used, thus not allowing the solution to be transferred to other similar systems. In this paper, we formulate a general framework that generates four distinct collision-free communication strategies to enable arm-arm coordination for C3DP using robotic manipulators. We considered collisions both between the arms with themselves and between the arms and the part being printed. The strategies are general in that they are agnostic to the number of printers, their kinematics, and their spatial configurations in the manufacturing environment. We conducted a study of the four strategies using a two-printer scenario and then physically validated them with four test cases of varying geometries. The results show that the strategies successfully produce printed parts while being collision-free. The makespan reduction using our strategies when compared to a single printer varied from 20% to 42% depending on the strategy used. Finally, we discuss the limitations of the framework, as well as future research directions.

Keywords: Cooperative 3D printing; Additive manufacturing; Industrial manipulators; Swarm manufacturing.

1 Introduction

In this paper, we formulate a collision-free communication framework to enable arm-arm coordination for cooperative 3D printing (C3DP) using robotic manipulators. The paradigm shift from traditional to swarm manufacturing has gained tremendous traction in recent years, as evidenced by the increasing number

*Address all correspondence to this author.

of publications on the topic [1–5]. C3DP, a necessary step to enable future large-scale swarm manufacturing, is a novel additive manufacturing technique that employs multiple mobile printers to achieve parallel processing in a given manufacturing task [6]. Unlike conventional gantry-based systems, C3DP is not limited in build volume, thus theoretically being able to print arbitrarily large objects. Furthermore, the simultaneous use of multiple extruders promises to significantly reduce the makespan of additive manufacturing tasks. However, the benefits of C3DP come at the cost of complex communication protocols between printers, which must be carefully coordinated to ensure efficient and collision-free operation.

A popular setup for C3DP is to use robotic manipulators with 3D-printing nozzles as the end-effector. Although several studies show functional and physically validated C3DP robotic arm systems, their collision avoidance strategies are generally specific to their own firmware, hardware, and manufacturing environment. They achieve collision-free printing either by generating their own custom G-code or making extensive changes to existing G-code produced by commercial software such as Cura or Prusa. This lack of flexibility hinders the dissemination and widespread use of C3DP among end-users who do not have the tools to perform these complex G-code manipulations to successfully produce a C3DP part.

2 Related Research

Although the number of publications focused on C3DP using manipulators has steadily increased over the years, there are still relatively few studies focused on formally addressing collision between printers during manufacturing.

Finding a collision-free path for cooperative robotic arms is not a problem singular to additive manufacturing. This research direction has been explored by multiple studies in other manufacturing applications. Larsen et al. designed a collision-free framework for cooperative assembly of aircraft fuselage using two industrial manipulators. The approach was based on the detection of collisions with bounding boxes and approximate cell decomposition [7]. Fei et al. developed a real-time, resolution-optimal path planner for a reconfigurable dual-arm robot, widely used in fast assembly applications [8]. The study employed a configuration space approach where the obstacle boundaries are decomposed. Foregoing searching for a global optimum in a high-dimensional space, Su et al. formulated a fast sampling-based technique to generate collision-free paths for two 6-DOF manipulators that satisfy real-time planning requirements [9].

However, C3DP has unique challenges to address beyond those pertaining to existing studies. In most additive manufacturing applications, the toolpath must be followed exactly, which means that many solutions found using the techniques described above would be invalid. Furthermore, existing studies offer feasible solutions at the cost of high computational resources, meaning that they cannot be used for real-time printing when more than two manipulators are active and moving continuously at high speeds. Hence, applying general robotic arm cooperation techniques, used in other manufacturing scenarios, to C3DP is not a straightforward task, since the nature of the processes is completely different.

A common approach to enable collision-free C3DP is to generate custom toolpaths. A collision-free toolpath in cooperative systems can be formulated as a co-scheduling constrained optimization problem. Jiang et al. generated toolpaths for a C3DP system using a modified evolutionary algorithm with embedded heuristics [10]. Jin et al. [11] solved the same problem, but with a greedy approach and different heuristics. However, these studies focused only on the collision between the nozzles and not the arms of the manipulators driving the end-effector. Another underlying assumption of these approaches is that the printers are always traveling at constant speed and the acceleration and deceleration limits are neglected. Furthermore, since the optimization problem proves to be NP-hard, the computational expenses become prohibitive as the problem scales up [12].

Another possibility is to modify a toolpath that was generated without taking cooperation into account. Shen et al. developed a successful four-printer C3DP system using triaxial robotic arms. This cooperative system reduced the make-span in large part by 73% [13]. For collision avoidance, they employed a technique based on safe and unsafe regions, similar to one algorithm we developed in this study, except that they modified the G-code extensively by reordering the sequence of commands. Furthermore, they restricted their printing process to be synchronous at the layer level. Krishnamurthy et al. developed a collision-free strategy for C3DP that combines safe region ideas with a moving front strategy, where the printers start closest to the middle of the printing area and move away from each other [14]. While this approach works well for a plethora of manufacturing scenarios, it relies on a pre-planned toolpath, which is calculated using complex meshing and interlocking geometry ideas.

Other hybrid approaches for C3DP have also been explored. Zhang et al. employed a simultaneous localization and mapping (SLAM) technique to devise a C3DP system using two concrete printers attached to a holonomic mobile base [15]. While the authors successfully printed a large concrete part concurrently, their collision-avoidance protocol is highly reliant on precise and frequently updating sensor data, as well as real-time, computationally expensive, trajectory re-planning requirements. In a study of multi-resolution AM, Bhatt et al. developed a methodology to produce collision-free paths for two 6-dof manipulators with respect to a part's planar and non-planar layers [16]. They discretize their manipulators' bodies and extruders using spheres. However, since the target print is small, they only use one robot at a time and do not consider robot-to-robot cooperation.

To summarize, there is a knowledge gap in the development of simple collision-free strategies for C3DP using robotic arms.

The solutions found in the literature are either only applicable to unrelated manufacturing scenarios, rely on complex custom toolpath generation, or perform extensive manipulation and reordering of existing G-code files to ensure that the printers will not collide, which mostly rely on pre-planning of the toolpaths. To the best of our knowledge, there has been no report to develop a general mathematical framework for real-time collision-free C3DP with robotic arms. Our work initiates research to address this knowledge gap by developing and systematically investigating such a framework. As such, our framework is general and scalable in that it is agnostic to the number of robotic manipulators, the kinematics of the manipulators, the spatial configuration of the manipulators, and hardware and firmware implementations. In order to demonstrate this framework, we focus our study on a two-robot system without loss of generality.

3 Problem Formulation

In this section, we formally define and formulate the problem of interest. We first introduce the notation used throughout the paper and then specify the scope of theoretical analysis and physical validation of the collision-free framework.

3.1 Preliminaries and Notation

We name the printers in an active C3DP process as P_i, where the subscript $i \in \mathbf{Z}^+$ denotes the printer number. The total number of printers in any process is given by $N \in \mathbf{Z}^+$, meaning that $i \leq N$. Each printer P_i has a corresponding frame of reference denoted by $\{F_i\}$. Let $\vec{\xi}_i \in \mathbf{S}_i$ be the pose of the end-effector of printer P_i with respect to its coordinate frame $\{F_i\}$, where $\mathbf{S}_i$ represents the printer's task space. Since we often have to represent all vectors $\vec{\xi}_i$ in a single spatial frame of reference $\{A\}$, we will use the convention that the frames $\{A\}$ and $\{F_1\}$ are aligned, meaning that all the end-effector poses are defined with respect to $\{F_1\}$. For instance, in order to represent $\vec{\xi}_i$ with respect to $\{F_1\}$ we would apply the following rigid-body transformation:

$$^{F_1}\vec{\xi}_i = \vec{p} + R(^{F_i}\vec{\xi}_i), \tag{1}$$

where $\vec{p}$ is the origin of the frame $\{F_i\}$ with respect to the frame $\{F_1\}$ and R is the rotation matrix representing the orientation of the frame $\{F_i\}$ with respect to the frame $\{F_1\}$. Additionally, we define the X, Y, and Z components of the end-effector of printer P_i as ξ_i^x, ξ_i^y, and ξ_i^z, respectively.

The projected area occupied by the body of a printer P_i, as seen from above, at any time when it is commanded to an arbitrary end-effector position $\vec{\xi}_i(t)$ is denoted by $R_i(t)$. Suppose that there are two successive end-effector positions $\vec{\xi}_i(t_0)$ and $\vec{\xi}_i(t_1)$. The swept area $S_i(t)$ is defined by successive projections $R_i(t)$ that are continuously created as the printer moves from the time step t_0 to t_1.

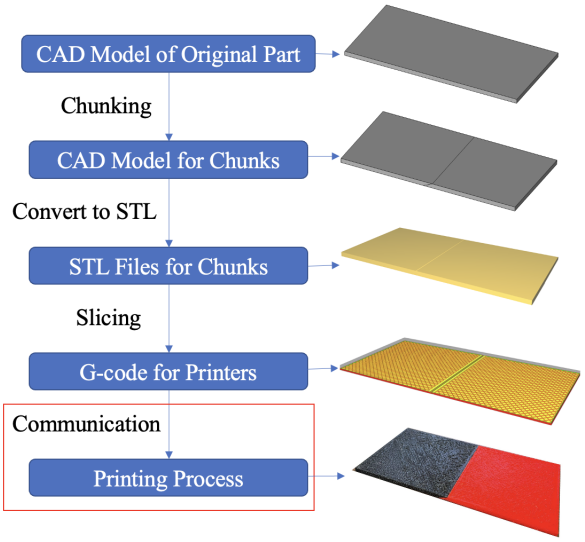


FIGURE 1: OVERVIEW OF C3DP PROCESS

3.2 Geometric Assumptions

In keeping with prior works on C3DP, we assume that the volume of a given part is split into mutually exclusive and exhaustive sub-volumes. These sub-volumes have been called *chunks* in early works in C3DP [17]. We also assume that no sub-volume is printed by more than one printer simultaneously. What is interesting to note is that the process of chunking in most C3DP workflows is incorporated directly within the slicing algorithms of a part. This necessitates development of specialized algorithms for G-code generation that are tailored so as to make chunk-based printing collision-free. In contrast, our method acts as a *plug-in* that can be applied at the end of any currently available slicing process (Fig. 1). Specifically, our workflow assumes that: (a) a part has been split into a desired number of chunks based on the number of printers and (b) each chunk is independently sliced with no considerations for collision-free path planning. As a direct consequence of this workflow, our framework does not need to impose any restrictions on how the part is divided into chunks.

3.3 Manufacturing Setup & Scope of Investigation

Given a set of robots, each printing a chunk at a certain location, the premise behind our framework is simple — to relay and receive minimal information (i.e. distance) across neighboring robots. The framework looks at different means (i.e. communication protocols) to achieve this simple objective. In principle, the core idea behind our framework is extendable to an

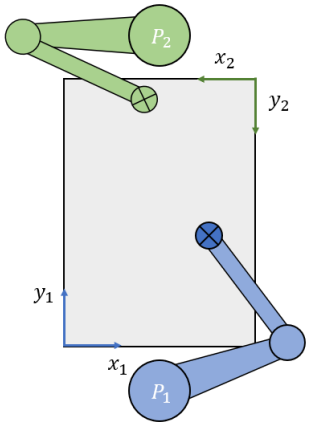


FIGURE 2: CASE STUDY USING SCARA PRINTERS

Cooperation in the Z-direction	Cooperation in the XY-directions: Safe Region	Cooperation in the XY-directions: Minimum Distance
Synchronized	Synchronized Safe Region	Synchronized Minimum Distance
Non-synchronized	Non-synchronized Safe Region	Non-synchronized Minimum Distance

FIGURE 3: COMMUNICATION FRAMEWORK MATRIX

arbitrary manufacturing setup defined by any number of printing robots (each with arbitrary kinematics) placed in arbitrary spatial configurations. Having said that, applying the framework is obviously not trivial for arbitrarily complex manufacturing setups. In fact, no single protocol may be applicable to every manufacturing setup. Therefore, in this work, our main goal is to systematically investigate each protocol through a series of concrete experiments using a two-robot manufacturing setup. In this setup (Fig. 2), we consider two printers with SCARA-type kinematics placed symmetrically across a rectangular print-bed [18]. Each printer has its own coordinate frame, and both share a finite rectangular printing area. This setup provides a simple yet representative case which is simultaneously amenable to both a computational analysis and an experimental validation.

4 Communication strategies

Based on whether robots need to be synchronized at the completion of each layer and the level of conservativeness in collision-free cooperation, communication strategies for C3DP can be categorized into a two-by-two matrix (Fig. 3). The four strategies are: (1) synchronized safe region, (2) synchronized minimum distance, (3) non-synchronized safe region, and (4) non-synchronized minimum distance. In what follows, we first discuss the difference between synchronized and non-synchronized communication in our context and then proceed to define the safe region and minimum distance strategies.

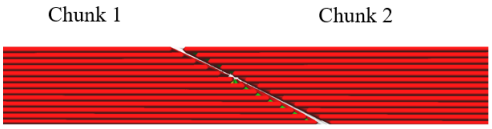


FIGURE 4: GEOMETRIC CHUNK CONSTRAINTS

4.1 Synchronized Communication

We use the term *synchronized* to indicate that the printers are constrained to working on the same layer at all times. Assuming the scenario with two printers, and leveraging the notation introduced in Section 3.1, we can formally define synchronized communication as:

$$\xi_1^z(t) - \xi_2^z(t) = 0 \tag{2}$$

In other words, if P_1 finishes printing its section of layer n, it must wait until P_2 also finishes its section of layer n before proceeding together to the subsequent layer $n+1$. This can be naturally extended to a multi-printer scenario. Therefore, a synchronized N-printer C3DP process must satisfy the following constraint:

$$\xi_i^z(t) - \xi_j^z(t) = 0, \ \forall i, j \leq N \tag{3}$$

The concept of synchronization is critical to C3DP since there will often be geometric constraints that need to be considered. For instance, two parts may be linked by a sloped interface (Fig. 4). If synchronization is not enforced and the printer responsible for Chunk 2 is more than one layer ahead of the printer responsible for Chunk 1, the interfacing section of Chunk 2 will restrict access to the other printer. This is closely related to the idea of geometric chunk dependency, which has been extensively studied in some of our previous work [19–21]. Furthermore, synchronization can be a simple yet effective way to address accumulated errors due to uncertainties.

4.2 Non-synchronized Communication

We use the term *non-synchronized* to indicate that the printers are not constrained to working on the same layer at all times. Recall that we operate under the assumption that all printers are printing layers of equal height. Let the chosen layer height for a process be $L_h \in \mathbf{R}^+$. We can relax the constraint imposed on the synchronized definition in Eqn. (2) and formally define non-

FIGURE 5: DETERMINING λ VALUE

synchronized communication for two printers as a set of constraints:

$$\xi_1^z(t) - \xi_2^z(t) \leq \lambda_{12} L_h, \tag{4}$$

$$\xi_2^z(t) - \xi_1^z(t) \leq \lambda_{21} L_h, \tag{5}$$

where λ_{12} is the maximum layer difference P_1 can be ahead of P_2 and λ_{21} is the same, but when P_2 ahead of P_1.

The reason why λ_{12} need not be equal to λ_{21} can be explained by using geometric chunk dependencies. Referring to Fig. 4, if Chunk 2 is even one layer ahead of Chunk 1, there will be a collision between the end-effector and the printed material, which means that λ_{21} must be zero. However, Chunk 1 is allowed to be ahead of Chunk 2, which means that λ_{12} will be a positive integer. Note that if $\lambda_{12} = \lambda_{21} = 0$, this set of constraints reduces to Eqn. (2), which should be expected since it implies that the layers are synchronized.

Using the same rationale as in the synchronized communication section, we can extend the non-synchronized definition to a multi-printer scenario. An non-synchronized N-printer C3DP process must satisfy the following constraint:

$$\xi_i^z(t) - \xi_j^z(t) \leq \lambda_{ij} L_h, \ \forall i, j \in N \tag{6}$$

In order to successfully allow non-synchronized communication in C3DP, one must carefully select the λ_{ij} parameter so that the manufacturing quality of the part is not negatively influenced. A good choice of λ_{ij} is directly dependent on the geometric constraints of the nozzle as well as the interfacing configurations, such as the slope of the interfacing angle. For the example shown in Fig. 5, where the chunks are separated by a sloped interface, it is immediately observed that λ_{21} should be zero since Chunk 2 is dependent on Chunk 1. However, to find a value for λ_{12} one needs to take into consideration the nozzle geometry. This has the direct implication that non-synchronized C3DP favors nozzles with sharper tips, resulting in smaller cross-sectional areas and allowing larger λ values.

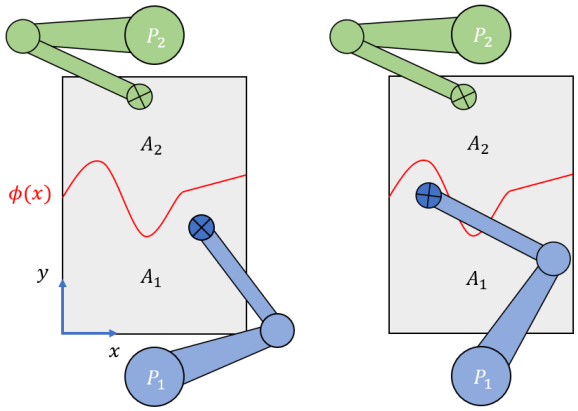


FIGURE 6: LEFT: P_1 IN A_1 ONLY. RIGHT: P_1 IN BOTH A_1 AND A_2

4.3 Safe Region

Our safe region communication strategy is developed based on a configuration space representation of the shared workspace of all printers involved in the C3DP process.

Consider the manufacturing scenario depicted in Fig. 6, and assume that the print consists of the entire shared rectangular area. Furthermore, assume that one of the layers of the print has an interfacing path defined by an arbitrary function $\phi(x)$. Following our convention, this function is defined with respect to the spatial frame chosen to align with $\{F_1\}$. Note that $\phi(x)$ divides the part into two regions, namely A_1 for printer P_1 and A_2 for printer P_2. However, even if P_1 is printing within the bounds of A_1, part of its body or end-effector could overlap with region A_2, which could potentially lead to a collision between the arms.

The main idea to circumvent this issue is to define two additional functions $\phi_1(x)$ and $\phi_2(x)$ that create two buffer regions B_1 and B_2, respectively (Fig. 7). The union of B_1 and B_2 is what we call the interfacing region, denoted by I:

$$I = B_1 \cup B_2 \tag{7}$$

Therefore, we can characterize an interfacing region to be safe if it satisfies the following set of constraint:

$$R_1(t) \notin A_2, \ \forall \vec{\xi}_1(t) \in \phi(x). \tag{8}$$

$$R_2(t) \notin A_1, \ \forall \vec{\xi}_2(t) \in \phi(x). \tag{9}$$

In other words, for all the end-effector positions of P_1 that trace the function $\phi(x)$, the printer's body cannot be

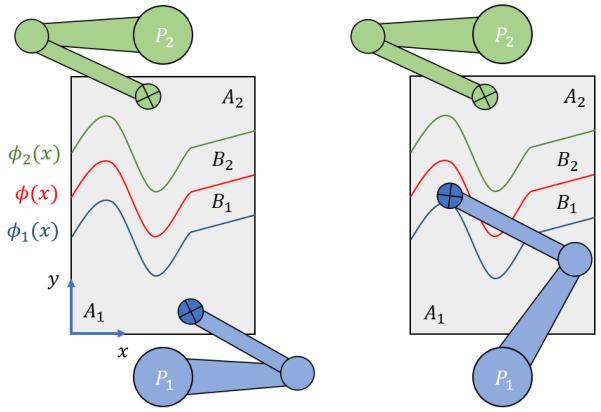


FIGURE 7: LEFT: P_1 IN A_1 ONLY. RIGHT: P_1 IN A_1, I, BUT NOT A_2

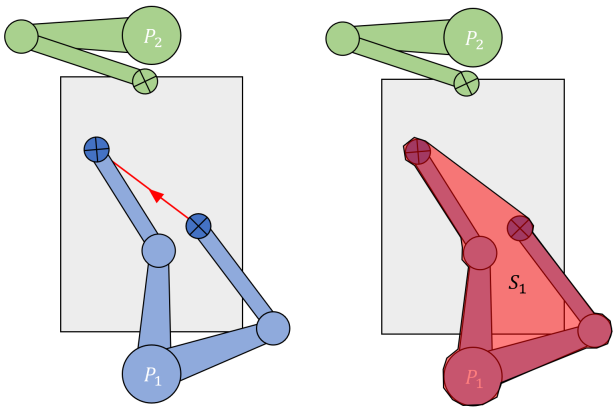


FIGURE 8: LEFT: P_1 UNDERGOING LINEAR MOTION. RIGHT: APPROXIMATE SWEPT AREA S_1 DUE TO LINEAR MOTION

in A_2. The same is true for P_2 with respect to A_1. The challenge becomes to find the functions $\phi_1(x)$ and $\phi_2(x)$ that create a safe interfacing region I while minimizing its size. This notion can be extended to the scenario with N printers, where there are $\{\phi_1(x), \phi_2(x), ..., \phi_N(x)\}$ functions defining $\{B_1(x), B_2(x), ..., B_N(x)\}$ regions. Hence, the constraints become:

$$R_i(t) \notin A_j,\ \forall \vec{\xi}_i(t) \in \phi(x),\ \forall i, j \leq N,\ i \neq j. \tag{10}$$

Having defined the steps necessary to generate a safe interfacing region I, and assuming the scenario with two printers, the safe region communication strategy must satisfy the following constraint:

$$\vec{\xi}_1(t) \in I \ \wedge\ \vec{\xi}_2(t) \notin I \tag{11}$$

This means that if the printer P_1 is currently in the interface region I at time t, the printer P_2 must be outside of it. This can naturally be extended to a multi-printer scenario. An N-printer C3DP process using the safe region as a communication strategy must satisfy the following constraint:

$$\vec{\xi}_i(t) \in I \ \wedge\ \vec{\xi}_j(t) \notin I,\ \forall i, j \leq N,\ i \neq j \tag{12}$$

Therefore, only one printer is allowed to be in the interfacing region at all times during the printing process. Although conservative, this strategy guarantees a collision-free C3DP process as long as functions $\phi_i(x)$ are chosen correctly.

4.4 Minimum Distance

Our minimum distance communication strategy aims to be less conservative than the safe region at the cost of added complexity, and thus more computational expenses. The key idea is to calculate the swept area $S_i(t)$, as defined in Section 3.1, at all times for each printer.

Consider the manufacturing scenario depicted in Fig. 8, the Printer P_1 is undergoing a linear G-code motion (either G0 or G1). Given the SCARA kinematics, which are well-defined for a right-armed printer, one can approximate the swept area S_1 during that motion as shown in the shaded area of Fig. 8. The same can be done for the printer P_2. Therefore, for this two-printer scenario, we can generate the following constraint:

$$S_1(t) \cap S_2(t) = \emptyset \tag{13}$$

It is generally a good idea to dilate the polygon created using the swept area method by a small value ε. This ensures that the arms do not just barely scrape each other. Therefore, the value of 2ε is what we define as the minimum distance that the arms are allowed to be between each other in any G-code motion. Finally, we can extend Eqn. (13) to a multi-printer scenario. An N-printer C3DP process using the minimum-distance communication strategy must satisfy the following constraint:

$$S_i(t) \cap S_j(t) = \emptyset,\ \forall i, j \leq N,\ i \neq j \tag{14}$$

This approach can be seen as a relaxed version of the safe region strategy. If we consider the safe interfacing region I described in Section 4.3, the printers would be able work on it simultaneously, using our minimum distance technique, as long as the intersection of their dilated swept areas is an empty set.

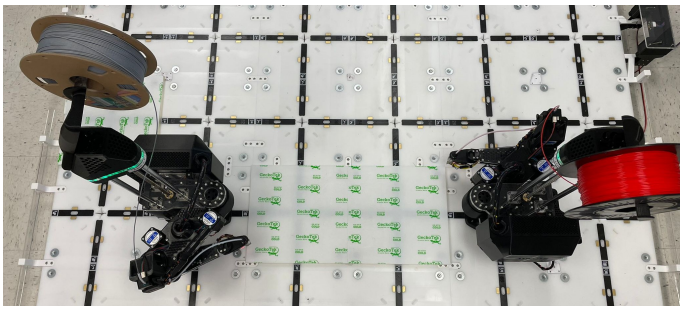


FIGURE 9: PHYSICAL VALIDATION SETUP

5 Physical Validation

In this section, we describe the process of physically validating the communication strategies. We outline the lab setup, the test cases, show an example of how to perform the validation step by step, and present the results obtained.

5.1 Experiment Setup

The setup in our laboratory involves a dual-printer C3DP scenario, mirroring the configuration we utilized in our theoretical analysis of the communication strategies. We used two SCARA printers developed by AMBOTS facing each other and sharing a rectangular printing area with $300mm \times 600mm$ dimensions (Fig. 9).

The printers are equipped with a Duet controller board running the RepRap firmware. They are both connected to a local network via WiFi, which is used to send and receive G-code commands. The only commands used for communication purposes are resume, stop, and request status, which provides the current G-code command that the printer is carrying out. At the higher level, we have a pure Python script running on a hub that maintains communication with the printers and oversees the printing process, and sends those G-code commands over HTTP as needed. There are no external sensors for real-time data transmission.

The Python program that oversees printing is different depending on the communication strategy used, but has a general algorithmic layout (Fig. 10). First, it initializes both printers and starts a loop that only terminates when they are idle. It continuously requests the printers' status at regular time intervals and then decides if it should stop or resume a printer, based on whether any constraints are being violated or if a printer is paused.

5.2 Test Cases

There are three important factors that we considered to design our test cases, namely: (1) the division of work (volume- and layer-level) between robots, (2) the geometry of the interfacing surface between two neighboring chunks, and (3) the geometry

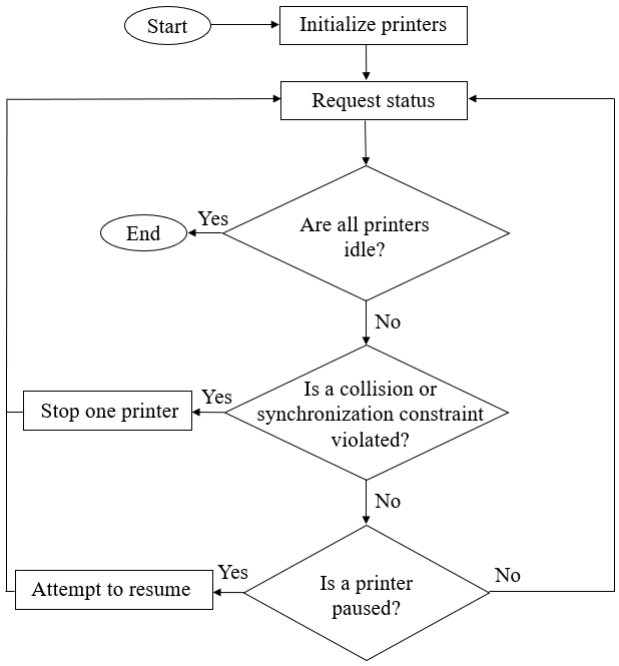


FIGURE 10: ALGORITHM FLOWCHART

of the interfacing curve at a given layer/slice between two neighboring chunks. In order to organize our test cases on the basis of these factors, we look at two specific works, namely the earlier *chunk-based printing* by McPherson et al. [17] and *LayerLock* by Krishnamurthy et al. [14]. Our rationale is as follows:

Division of work: Both these works divide the part volume in a manner that each robot ends up printing an equal volume. However, this equal volume distribution is implicit in *chunk-based printing* whereas *LayerLock* explicitly prescribes equal distribution of work not only for the chunk volume but even at each layer (whenever possible). Our goal is to demonstrate and compare both equal and non-equal volumetric as well as layer-wise division.

Interfacing surface geometry: This is where *chunk-based* and *LayerLock* differ significantly. The *chunk-based* approach subscribes to a simple planar interface between two chunks with a nearly sequential degree of cooperation — one robot prints atop the interface after the first one has finished printing the interfacing surfaces of its respective chunk. *LayerLock*, on the other hand, is a purely-geometry-based approach resulting in interlocking interfaces with a highly synchronized layer-wise cooperation strategy. As a result, the application of standard slicing workflow is not possible for *LayerLock*. Therefore, we choose the middle-ground wherein we seek to develop test cases with simpler planar interfaces (*chunk-based*) with the possibility to implement layer-wise cooperation (*LayerLock*).

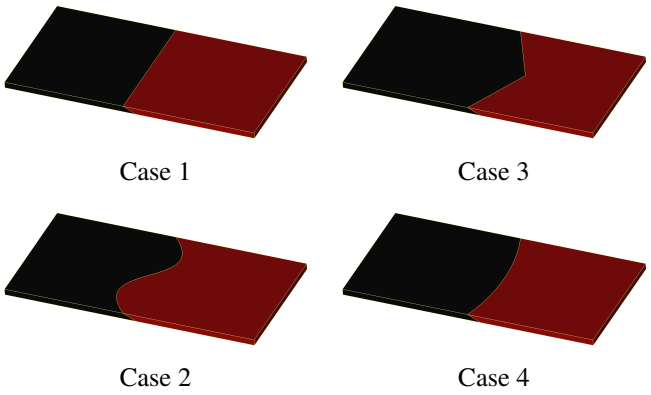


FIGURE 11: TEST CASES

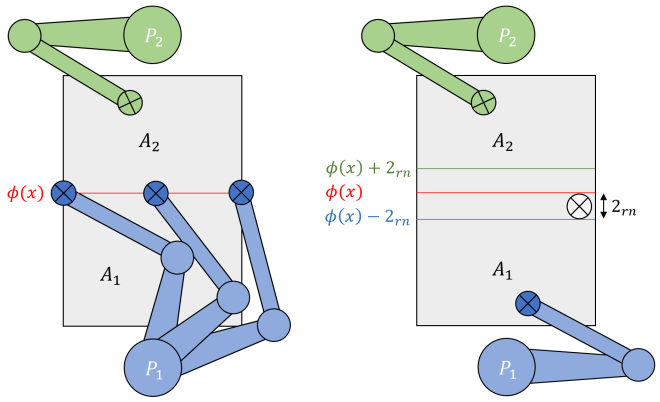


FIGURE 12: LEFT: ONLY NOZZLE OF P_1 ENTERS AREA A_2. RIGHT: OPTIMAL FUNCTIONS $\phi_i(x)$ ARE FOUND BASED ON NOZZLE RADIUS

Interfacing curve per-layer: Here, the interfacing curves in the *chunk-based* approach are simply straight lines, while *LayerLock* exploits a specialized cellular decomposition resulting in complex piece-wise linear paths at each layer based on the layer-mesh topology. Our approach is to consider interfacing curves that demonstrate generality from a fundamental geometric perspective. Therefore, we consider linear, piece-wise linear, and curved cases. Furthermore, our curves are also organized in terms of convexity and inflexion points. Incidentally, this organization also naturally lends itself to equal and non-equal division of labor at both layer and volumetric levels across different cases.

Based on the rationale above, we designed four case studies for the physical validation of our framework. As displayed in Fig. 11, they can be classified as follows: (1) equal volume straight interface, (2) equal volume curved interface, (3) unequal volume straight interface, and (4) unequal volume curved interface.

The validation of test case 1 was conducted with all four communication strategies and is described in detail in the next section. Test case 3 was validated using the synchronized safe region strategy. Test cases 2 & 4, on the other hand, were validated using the non-synchronized minimum distance strategy because finding good functions $\phi_i(x)$ for curved interfaces is non-trivial. Each test case has 10 layers, but for validation, we only printed the first 2 layers of test cases 2-4. The number of layers printed for each test case does not affect its validation, since all layers have the same interfacing path geometry, albeit shifted by a small amount due to the sloped interface and print uncertainties. However, to demonstrate the mechanical integrity of the sloped interface between the chunks, test case 1 was printed in its entirety.

5.3 Validation Example

Taking test case 1, equal volume straight interface, as an example, we now demonstrate how to apply the methodology developed in Section 4 in order to successfully produce a cooperatively 3D printed part.

The simplest way to enforce synchronization is to include layer number information in the G-code as a comment, and then use a message command to report that information to the host. If the commercial slicer software being used does not include that information when generating the G-code files, one can write a postprocessing script to include messages in them whenever the Z-component of any G-code command changes. It is not recommended to track the end-effector position to enforce synchronization, as pause commands will move the printer, changing the Z-component of the end-effector, and erroneously signaling to the other printers that synchronization is being violated.

Non-synchronized communication follows the same idea for keeping track of the current layer that each printer is printing. The only difference is in the enforced constraint, which uses precalculated λ values to delegate pauses and resumptions. Test case 1 has a sloped interface with an angle of approximately 26.57^o with respect to the horizon, whereas the angle of the nozzle tip of the SCARA printers is 83.23^o. Hence, we can quickly determine that $\lambda_{12} = 10$ (i.e., the maximum number of layers) and $\lambda_{21} = 0$, respectively.

It is important to note that no new commands are added or reordered in the G-code files. The modifications made are simply comments to allow for quick communication and information transfer between the host and the printers. This is a direct consequence of our black-box approach where we do not make toolpath changes or split the G-code of a chunk into multiple files.

In order to enable the safe region communication strategy,

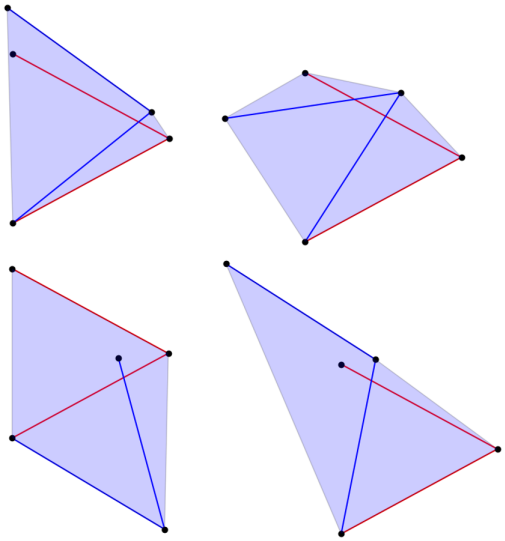


FIGURE 13: CONVEX HULL APPROXIMATION OF SCARA SWEPT AREA FOR DIFFERENT DESIRED CONFIGURATIONS (BLUE) STARTING AT THE SAME INITIAL CONFIGURATION (RED)

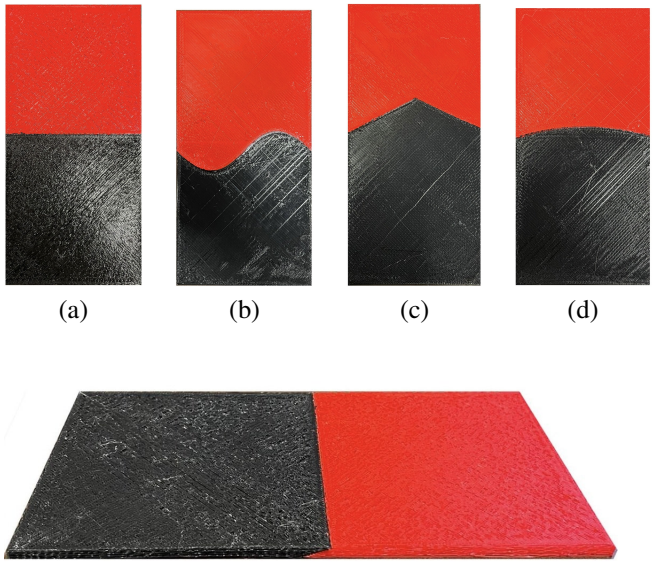


FIGURE 14: PRINTED CASE STUDIES. (A) CASE 1, (B) CASE 2, (C) CASE 3, (D) CASE 4, (E) FULL GEOMETRY FOR CASE 1 HIGHLIGHTING BONDED INTERFACE

we must find two functions $\phi_1(x)$ and $\phi_2(x)$ that define buffer zones B_1 and B_2, such that the generated interfacing region I is safe. The process of finding optimal functions $\phi_i(x)$ that minimize the interfacing area is not a trivial task, and we discuss this further in Section 6.3. For the purpose of validating test case 1, we first recognize that, due to the simplicity of the interfacing path and the SCARA kinematics of the printer P_1, all the end-effector positions $\xi_1(t)$ that trace $\phi(x)$ result in only the nozzle of the printer being in the region A_2. This is also true for the printer P_2 with respect to area A_1. This means that we can quickly find the optimal functions $\phi_1(x) = \phi(x) - 2r_n$ and $\phi_2(x) = \phi(x) + 2r_n$, where r_n is the nozzle radius (Fig. 12). After defining all regions, we can then add message commands to the G-code files, signaling whether or not a printer is in the interfacing region.

Finally, to employ our minimum-distance communication strategy, we need to find an accurate and computationally efficient way to calculate the robot's swept area $S_i(t)$. A reasonable approach for this scenario, in which SCARA kinematics apply, is to treat the problem as a convex hull calculation problem. From the G-code file, we have, at all times, access to the printer's most recent end-effector position and its desired end-effector position. Therefore, we have a total of five control points that we can use to approximate the swept area. Finding the convex hull of these five control points provides us with a convex polygon that represents the printer's swept area during that motion (Fig. 13).

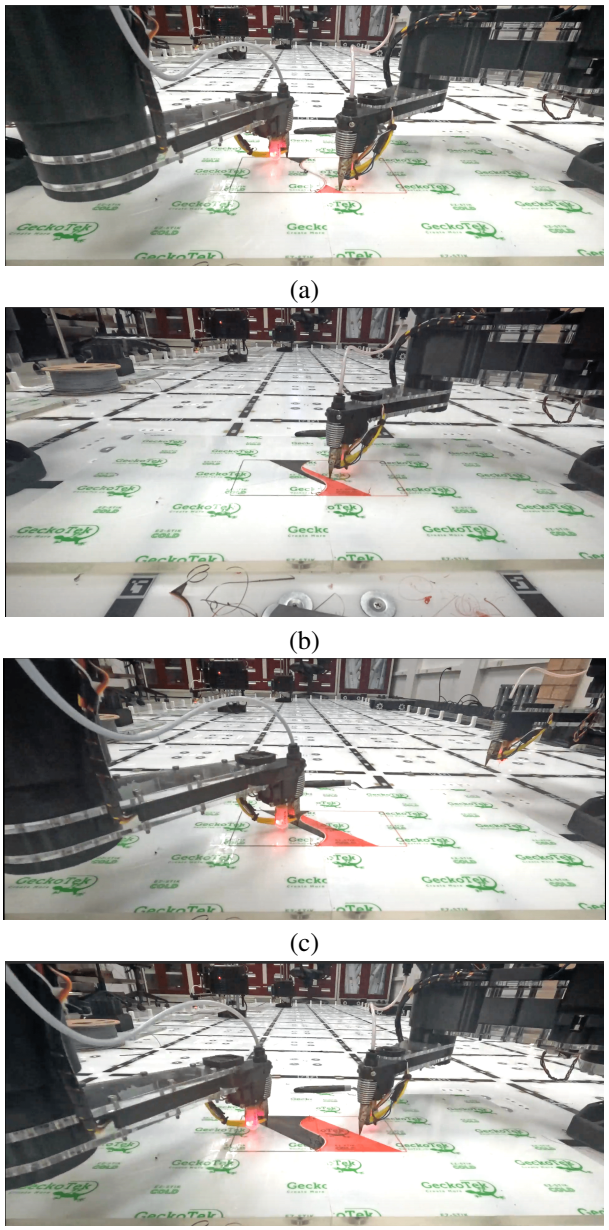
Since the number of control points is small, any popular algorithm for computing the convex hull is applicable here, and so we implemented Graham's scan [22]. Note that for higher degree of freedom manipulators, different techniques for computing the swept area (or volume) must be considered [23]. Hence, as long as the swept volume can be efficiently computed, our methodology remains agnostic to the type of manipulator kinematics used.

5.4 Results

We successfully used the framework to cooperatively print the four test cases without collisions (Fig. 14). As previously mentioned, only test case 1, was printed in its entirety, and its complete geometry is shown, highlighting the sloped connection between the chunks and how the mechanical bond of the interface is preserved in the cooperative process.

The makespan of the complete print of test case 1 using the safe region method was 176 minutes, and was 126 minutes when using the minimum distance method, compared to the single print makespan of 220 minutes. These values correspond to approximately 20% and 42% reduction in the makespan, respectively. This difference is consistent with the level of conservativeness of each algorithm, as outlined in Section 4.

We did not observe a noticeable difference between the synchronized and non-synchronized versions of these algorithms. This is mostly due to the small height of the test cases, which diminishes the impact of cooperation along the Z-direction. Finally, a time sequence of the cooperation procedure to print the first layer of test case 2, using non-synchronized minimum distance, is also shown in Fig. 15. A time-lapse video is also avail-



(a) (b) (c) (d) PRINTERS CONTINUE SIMULTANEOUS PRINTING

FIGURE 15: SNAPSHOTS OF VALIDATION OF CASE 2. (A) PRINTERS INITIALLY ACTIVE, (B) & (C) PRINTERS ALTERNATE TO AVOID COLLISION, (D) PRINTERS CONTINUE SIMULTANEOUS PRINTING.

able showing the cooperative process [1].

[1] https://tinyurl.com/2s48xhxy

6 Discussion

In this section we discuss the key insights generated from the results section. First, we analyze the effects of pausing the printers during the cooperative process. Pauses are evidently not optimal, but they are expected since the algorithm is general and uses a black-box approach, with no modifications to the toolpath. We then assess how the size of the print might affect the utility of C3DP. Finally, we further discuss the effects of manipulator kinematics and how they may affect scalability in multi-printer scenarios.

6.1 Effects of Pausing

In all algorithms within our collision-free framework, there are situations in which resuming the printing process for one printer will instantaneously cause the other printer to pause. The transition time between when one printer stops and when the other printer's nozzle reaches its goal position to resume printing is undesirable, since there is no forward manufacturing progress. Although this transition time is usually small, i.e. on the order of a couple of seconds, it is not negligible since the number of such stop procedures can be on the order of hundreds even for small prints, as demonstrated by our physical validation data: printing 10 layers of test case 1 using safe region led to 42 stop procedures. Defining this transition time as Δt and the total number of stop procedures as n_s, we can quantify the total wasted time T_w as:

$$T_w = n_s \Delta t \tag{15}$$

It is important to note that if T_w is large, it is possible that the C3DP process may yield a very small reduction in the makespan when compared to a single printer, which negates the value of using a cooperative framework. This can be seen by a greedy variant of our non-synchronized safe-zone implementation, which has a makespan of 206 minutes, compared to the estimated makespan of 220 minutes using a single printer. The difference between the greedy and the normal variant of our strategies is that the printers attempt to resume regardless, if that action will cause a collision-constraint, making the other printer pause instead. Therefore, minimizing T_w is an essential task for future practical applications of C3DP.

In minimizing T_w, especially from the perspective of this paper, where we do not pre-optimize the G-code path, the variable of interest is predominantly Δt over n_s. Note that:

$$\lim_{\Delta t \to 0} T_w = 0, \tag{16}$$

which, more importantly, implies that:

$$\Delta t = 0 \implies C_{double} \leq C_{single}, \tag{17}$$

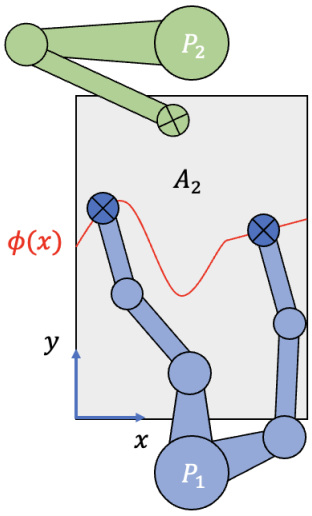


FIGURE 16: KINEMATICALLY REDUNDANT MANIPULATOR TRACING INTERFACING PATH

where C_{double} and C_{single} represent the makespan for a C3DP process using two and one printer, respectively. This means that we can guarantee that the makespan of any C3DP process will be no greater than its equivalent single-printer makespan, irrespective of pathing and collision-free algorithm choices, if the transition time is zero. The process of efficiently minimizing Δt, especially for multi-printer scenarios, merits a separate investigation reserved for future work.

6.2 Effects of Print Size

The effectiveness of C3DP in reducing the makespan of the single printer case may depend on the size of the print. By denoting the areas for which each printer is responsible as A_1 and A_2, and the interfacing region as I, following notation consistent with Section 4.3, we can define the complement of the interfacing region as:

$$I^c = A_1 \cup A_2 \tag{18}$$

The size of I^c has a direct impact on the number of pauses, which are undesirable, as discussed in Section 6.1. Because I^c depends on the geometry of the part, it may increase, decrease, or remain unchanged as the print size varies. However, generally we can claim that a C3DP process will have a greater impact in reducing the makespan of a single printer process when the ratio of the interfacing area and its complement becomes smaller. Extending the definition of I^c to a N-printer scenario gives:

$$I^c = \bigcup_{i=1}^{N} A_i \tag{19}$$

We can infer that as the ratio of the interfacing area and its complement (which we define to be k_A) tends to zero, the ratio of the makespan of the single printer scenario and the N-printer scenario C_N (being consistent with the notation introduced in Section 6.1) tends to N:

$$k_A = \frac{I}{I^c} \to 0 \implies \frac{C_{single}}{C_N} \to N \tag{20}$$

The ratio k_A could serve as a metric to guide us in deciding when a C3DP process is warranted. For instance, if k_A is greater than one, meaning that the interfacing area is greater than the non-interfacing area, our approach to C3DP may not be ideal, as there will be too many pauses. Conversely, if k_A is smaller than one, then our approach is more effective.

6.3 Effects of Arm Kinematics

The printers' kinematics are taken into account in all our algorithms. In the safe region strategy, functions $\phi_i(x)$ are chosen on the basis of kinematic constraints. In minimum distance, the swept area is directly dependent on the type of kinematic chain employed.

We noticed that there may be kinematics constraints that are more or less appropriate for C3DP. For instance, kinematics constraints that allow smaller interfacing regions and smaller swept areas are desirable. Hence, we argue that manipulators that are kinematically redundant in C3DP applications could be advantageous in reducing the number of required pauses.

Consider, for instance, how a kinematically redundant triaxial planar arm can trace the interfacing path $\phi(x)$ while keeping the arm orthogonal to it (Fig. 16). This type of redundancy resolution allows us to pick optimal functions $\phi_1(x) = \phi(x) - 2r_n$ and $\phi_2(x) = \phi(x) + 2r_n$ even in a curved interface, generating the smallest possible interfacing region where only the nozzle needs to be considered. Furthermore, if the minimum distance technique is used, kinematic redundancy allows different choice of movement to the same desired position, providing a range of possible swept areas. We could then choose swept areas that do not violate the algorithmic constraint and reduce the number of necessary pauses.

6.4 Scalability

Assuming the division of labor among multiple printers is given, the communication strategies developed in this paper are

scalable to multi-printer scenarios. All the constraints and algorithms presented are defined in such a way that arbitrary numbers of printers can participate in the cooperative process. However, computational cost can be a limiting factor to the scalability of our framework, depending on the strategy used.

If the safe region strategy is employed, there are no concerns regarding real-time computational cost, since the interfacing region is pre-calculated, and the constraint check is done in $O(1)$ access time. However, for the minimum distance strategy, the number of degrees of freedom of the manipulator has a significant impact on the real-time computational cost of finding the swept volume. This was not a concern in our work since we employed SCARA printers and showed that the swept area calculation could be translated into a planar convex hull problem with $O(n\ log\ n)$ time complexity and a low number of data points. For manipulators with higher degrees of freedom, exact computations of the swept volume are intractable for real-time applications. However, if we relax the need for exact swept volume calculations and approximate the manipulator's body with polyhedra, a number of available algorithms in literature are able to perform efficient computations [24]. Another possibility is to use AI techniques, such as deep neural networks, to learn the kinematics of any manipulator off-line and then use the trained model to quickly find real-time swept volume approximations [25].

7 Conclusion

In this paper, we successfully developed and physically validated a collision-free communication framework between printers in a C3DP process. The framework generated four distinct communication strategies, which are general and, therefore, can be implemented in arbitrary C3DP systems. Our philosophy was to approach the problem from a black-box perspective, where the only assumption from a C3DP pipeline standpoint is that the given G-code files are valid. The results showed that the framework is promising, and we look forward to exploring the research directions stemming from the challenges and insights presented in the discussion section.

ACKNOWLEDGMENT

The authors would like to thank Daniel Weber for his assistance in the physical validation of the communication framework and Matthew Ebert for his contribution in the design of the test cases.

REFERENCES

[1] Zhang, K., Chermprayong, P., Xiao, F., Tzoumanikas, D., Dams, B., Kay, S., Kocer, B. B., Burns, A., Orr, L., Choi, C., et al., 2022. "Aerial additive manufacturing with multiple autonomous robots". *Nature,* ***609***(7928), pp. 709–717.

[2] Auricchio, F., 2022. "A continuous model for the simulation of manufacturing swarm robotics". *Comput. Mech.,* ***70***(1), jul, p. 155–162.

[3] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O. L., Sha, Z., and Zhou, W., 2022. "Toward swarm manufacturing: Architecting a cooperative 3d printing system". *Journal of Manufacturing Science and Engineering,* ***144***(8), p. 081004.

[4] Park, H.-S., and Tran, N.-H., 2012. "An autonomous manufacturing system based on swarm of cognitive agents". *Journal of Manufacturing Systems,* ***31***(3), pp. 337–348.

[5] Oxman, N., Duro-Royo, J., Keating, S., Peters, B., and Tsai, E., 2014. "Towards robotic swarm printing". *Architectural Design,* ***84***(3), pp. 108–115.

[6] Alhijaily, A., Kilic, Z. M., and Bartolo, A. N. P., 2023. "Teams of robots in additive manufacturing: a review". *Virtual and Physical Prototyping,* ***18***(1), p. e2162929.

[7] Larsen, L., Pham, V.-L., Kim, J., and Kupke, M., 2015. "Collision-free path planning of industrial cooperating robots for aircraft fuselage production". *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2042–2047.

[8] Fei, Y., Fuqiang, D., and Xifang, Z., 2004. "Collision-free motion planning of dual-arm reconfigurable robots". *Robotics and Computer-Integrated Manufacturing,* ***20***(4), pp. 351–357.

[9] Chang, s., and Xu, J., 2022. "A novel non-collision path planning strategy for multi-manipulator cooperative manufacturing systems". *The International Journal of Advanced Manufacturing Technology,* ***120***, 05, pp. 1–26.

[10] Jiang, Z., Wang, H., and Sun, Y., 2021. "Improved co-scheduling of multi-layer printing path scanning for collaborative additive manufacturing". *IISE Transactions,* ***53***(9), pp. 960–973.

[11] Jin, Y., Pierson, H. A., and Liao, H., 2019. "Toolpath allocation and scheduling for concurrent fused filament fabrication with multiple extruders". *IISE Transactions,* ***51***(2), pp. 192–208.

[12] Applegate, D. L., Bixby, R. E., Chvátal, V., and Cook, W. J., 2007. *The Traveling Salesman Problem*. Princeton University Press, Princeton.

[13] Shen, H., Pan, L., and Qian, J., 2019. "Research on large-scale additive manufacturing based on multi-robot collaboration technology". *Additive Manufacturing,* ***30***, p. 100906.

[14] Krishnamurthy, V., Poudel, L., Ebert, M., Weber, D. H., Wu, R., Zhou, W., Akleman, E., and Sha, Z., 2022. "Layerlock: Layer-wise collision-free multi-robot additive manufacturing using topologically interlocked space-filling shapes". *Computer-Aided Design,* ***152***, p. 103392.

[15] Zhang, X., Li, M., Lim, J. H., Weng, Y., Tay, Y. W. D., Pham, H., and Pham, Q.-C., 2018. "Large-scale 3d printing

by a team of mobile robots". *Automation in Construction,* ***95***, pp. 98–106.

[16] Bhatt, P. M., Kabir, A. M., Malhan, R. K., Shah, B., Shembekar, A. V., Yoon, Y. J., and Gupta, S. K., 2019. "A robotic cell for multi-resolution additive manufacturing". In 2019 International Conference on Robotics and Automation (ICRA), pp. 2800–2807.

[17] McPherson, J., and Zhou, W., 2018. "A chunk-based slicer for cooperative 3d printing". *Rapid Prototyping Journal*.

[18] Hyden, Z., 2019. "Design of a scara based mobile 3d printing platform".

[19] Poudel, L., Zhou, W., and Sha, Z., 2021. "Resource-Constrained Scheduling for Multi-Robot Cooperative Three-Dimensional Printing". *Journal of Mechanical Design,* ***143***(7), 04. 072002.

[20] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O. L., Moquin, S. J., Sha, Z., and Zhou, W., 2020. "Architecting the cooperative 3d printing system". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 83983, American Society of Mechanical Engineers, p. V009T09A029.

[21] Weber, D. H., Zhou, W., and Sha, Z., 2022. "Z-chunking for cooperative 3d printing of large and tall objects". *Solid Freeform Fabrication 2022: Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium*.

[22] Graham, R. L., 1972. "An efficient algorithm for determining the convex hull of a finite planar set". *Info. Proc. Lett.,* ***1***, pp. 132–133.

[23] Täubig, H., Bäuml, B., and Frese, U., 2011. "Real-time swept volume and distance computation for self collision detection". In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 1585–1592.

[24] Kim, Y. J., Varadhan, G., Lin, M. C., and Manocha, D., 2003. "Fast swept volume approximation of complex polyhedral models". In Proceedings of the eighth ACM symposium on Solid modeling and applications, pp. 11–22.

[25] Chiang, H.-T. L., Baxter, J. E., Sugaya, S., Yousefi, M. R., Faust, A., and Tapia, L., 2021. "Fast deep swept volume estimator". *The International Journal of Robotics Research,* ***40***(10-11), pp. 1068–1086.