

# TOWARDS SWARM MANUFACTURING: ARCHITECTING A COOPERATIVE 3D PRINTING SYSTEM

Laxmi Poudel<sup>1</sup>, Lucas Galvan Marques<sup>2</sup>, Robert Austin Williams<sup>2</sup>, Zachary Hyden<sup>2</sup>, Pablo Guerra<sup>2</sup>,  
Oliver Luke Fowler<sup>2</sup>, Stephen Joe Moquin<sup>2</sup>, Zhenghui Sha<sup>3</sup>, Wenchao Zhou<sup>\*4</sup>

<sup>1</sup>Department of Mechanical Engineering  
University of Michigan, Ann Arbor, MI 48109, USA

<sup>3</sup>Walker Department of Mechanical Engineering  
University of Texas, Austin, TX 78712, USA

<sup>2</sup>AMBOTS Inc., Fayetteville, AR 72701, USA

<sup>4</sup>Department of Mechanical Engineering  
University of Arkansas, Fayetteville, AR 72701, USA

## ABSTRACT

*Swarm manufacturing is a new paradigm for distributed manufacturing, where each factory employs an army of mobile manufacturing robots working together to manufacture desired products. It focuses on building factories within a local supply chain, where multiple products can be produced to meet changing customer demand from mass manufacturing to on-demand production of customized products. Cooperative 3D printing (C3DP) is a primitive form of swarm manufacturing, where multiple mobile 3D printing robots work together cooperatively to print the desired part, representing a major step towards enabling swarm manufacturing. At the core of C3DP lies the chunk-based printing strategy. This strategy splits the desired part into small chunks. The chunks are then assigned and scheduled to be printed by individual printing robots. In our previous work, we presented various hardware and software components of C3DP, such as mobile 3D printers, chunk-based slicing, scheduling, and simulation. This paper presents a fully integrated and functional C3DP platform with all necessary components and outlines how they work in unison from a system-level perspective. Specifically, the new architecture of hardware and software includes a set of new chunking strategies, a scalable scheduler for multi-robot printing, a SCARA-based printing robot, a mobile platform for robot transportation, a system of modular floor tiles, and a charging station for the mobile platform. Finally, we demonstrate the capability of the system using two case studies with successful large-scale prints.*

**Keywords:** Cooperative 3D printing, Swarm 3D Printing, Multi-robot 3D printing, Swarm Manufacturing

## 1. INTRODUCTION

While they are increasingly connected and automated, modern factories are still largely designed based on the philosophy of assembly line since it was first implemented by Henry Ford in 1913. Over the past decades, manufacturing systems have evolved significantly to meet several critical requirements, including shorter lead time, greater varieties, lower prices, and flexible production to accommodate fluctuating market demands [1–4]. These demands have led to new manufacturing paradigms, such as flexible manufacturing systems (FMS) [5], reconfigurable manufacturing systems (RMS) [6,7], smart manufacturing [8,9], cloud-based manufacturing (CBMS) [10,11], etc. While these manufacturing paradigms have greatly reshaped modern factories, the factories are still primarily designed to produce a specific product or a few specific products for an extended period of time. While the factories may be reconfigured to produce a different product, it often takes a significant amount of time and manual work to recalibrate the machines in the factory. Therefore, its configuration cannot typically be changed during its production time. For a specific configuration, the factory is normally designed with three constant parameters: product dimensions, production capacity, and a set of needed manufacturing processes. Although this works well for an infrequent change of production demands, it relies on the economy of scale to distribute the overhead cost of reconfiguration to many units and lower the average manufacturing cost per unit, which invariably leads to concentration of manufacturing and potential congestion and disruption in the supply chain. This is particularly true when some critical factories or transportation routes in the supply chain are forced to shut down during a regional or global crisis like COVID-19.

Swarm manufacturing is a new paradigm for distributed manufacturing, where each factory employs thousands of mobile manufacturing robots that can regroup and recalibrate themselves in real-time. In this paradigm, the product dimensions, the production capacity, and the set of manufacturing processes, which typically remain unchanged in modern factories for a specific production configuration, can all be changed in real-time by adding more functional robots or regrouping the robots with designed software. Due to the real-time reconfiguration of the manufacturing robots, the production overhead is negligible, and thus the average manufacturing cost per unit is relatively insensitive to the production quantity. The production scale can be achieved by distributing them across a global network of thousands of swarm-manufacturing-powered factories based on regional demands.

While being a promising alternative to existing manufacturing paradigms, swarm manufacturing has not been realized so far. In our prior work, we developed various elements for cooperative 3D printing (C3DP), a primitive form of swarm manufacturing, where the

---

\* Corresponding authors: [zhouw@uark.edu](mailto:zhouw@uark.edu)

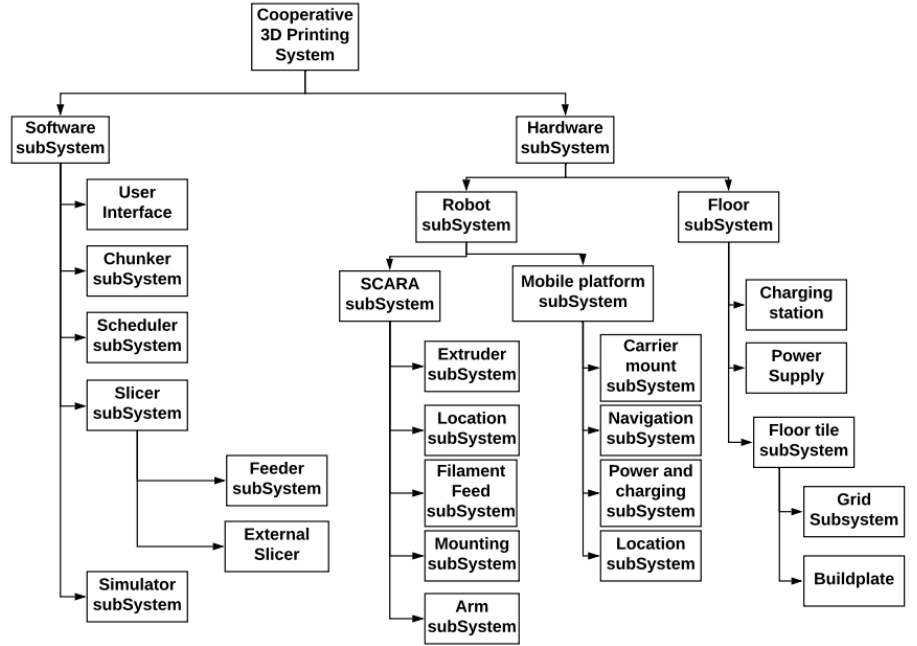
manufacturing robots are primarily 3D printing robots. The use of 3D printing or additive manufacturing allows manufacturing of wide variety of complex products directly using a CAD model [12]. In C3DP, multiple mobile 3D printing robots are simultaneously activated to print large-scale objects in parallel. To enable cooperation among multiple robots, a part is first divided into small chunks using a sloped chunking strategy; then, these chunks are assigned to multiple robots that work parallelly to fabricate the allocated chunks. Since each chunk has a sloped surface, the new material gets deposited on the sloped surface. The bonding between chunks will be naturally formed following the same bonding mechanism of the traditional layer-based printing. This new approach to additive manufacturing breaks the conventional gantry-based AM that has a limitation in scalability and speed.

The objective of this paper is to present the realization of the first C3DP system as a fundamental step towards swarm manufacturing, including both hardware and software architecture, as shown in **Figure 1**. In this paper, we present how different components in the system interact with each other and the role of each component as a part of the entire system. The major contributions of the study are summarized in three aspects:

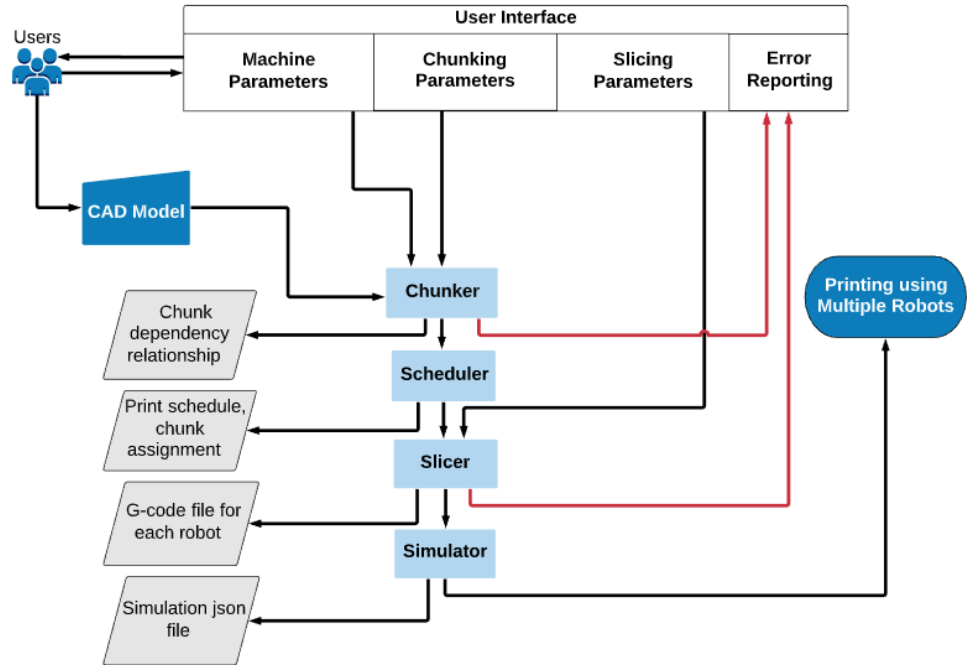
1. The architecture and integration of the software and hardware that form a fully functional C3DP system for the first demonstration of the swarm manufacturing concept.
2. An improved C3DP software system that offers different chunking options, improved slicing operations, and a more robust C3DP simulator.
3. An upgraded C3DP hardware system which includes a new SCARA-arm printer as well as a new modular floor system.

The paper is organized as follows. In **Section 2**, a review of the related research is presented followed by the architecture of C3DP in **Section 3**, focusing on the software components and how they are inter-connected. The new generation of the hardware platform is presented in **Section 4**. In **Section 5**, an introduction of the system, including both hardware and the software, is provided, along with its implementation and validation. The validation is achieved by both simulation and physical implementation of C3DP in two case studies. Finally, the conclusions and the future work are presented in **Section 6**.

## 2. RELATED RESEARCH



**Figure 1.** The Physical system architecture of Cooperative 3D printing system



**Figure 2:** Process flow in Cooperative 3D printing from high level perspective.

Although studies on swarm manufacturing are scarce, research related to large scale additive manufacturing has been on a steady rise over last few years [13–16]. Alternate 3D printing systems and methods have been developed for increasing the print speed and the scale of the print object, such as Big Area Additive Manufacturing (BAAM) [17] and concurrent printing that uses multiple extruders [18]. In these systems, increasing the scale of an object requires increasing the size of the printer itself, and increasing the print speed requires the sacrifice of printing resolution by using a larger print nozzle (e.g., BAAM). The use of multiple robots to print a single object improves the print speed as well as the scale without increasing the size of the printers themselves. This is because the size of the print object is no longer guided by the size of the printer used to print the object, and the multiple extruder-carrying robots can work together to minimize the total print time. The increasing number of ongoing research in cooperative additive manufacturing demonstrates the potential benefit of such systems. Hongyao et al. demonstrated the use of multiple robots in printing applications collaboratively by using four large-scale robots to print an object [19]. They reported an increase in the system’s efficiency by more than 73% when four robots were used compared to the traditional single extruder gantry-based 3D printing system. However, the system used static robots, and thus, the scale can only be increased in one direction. Similarly, a group from MIT presented Robotic Swarm Printing (RSP) that used a swarm of printers for large-scale construction [20]. However, the presented system can only print tubular structures, and it is not clear how the approach translates to printing different geometries. A study at Singapore’s Nanyang Technological University revealed large-scale printers that can print concrete structures concurrently using two mobile robots with Simultaneous Localization and Mapping (SLAM) [21]. Such collaborative printing using mobile robots requires additional considerations for path planning and slicing [22] that would otherwise not be needed, for static gantry-based printing system. To make improvement on additively manufactured concrete concrete surfaces, Hack et al. presented a combination of additive fabrication and formative and subtractive postprocessing, which was demonstrated by fabrication of  $2.5\text{ m} \times 2.3\text{ m} \times 0.18\text{ m}$  concrete wall [23]. Similarly, an ongoing study at the New York University is developing a framework for collective additive manufacturing robots that can autonomously cooperate while printing large structures [24].

While the research area is gaining attention, the foundation to test new research ideas and support multi-robot cooperative 3D printing is lacking. There are no existing integrated hardware and software frameworks that provide a streamlined testbed and manufacturing foundation for cooperative 3D printing. This study aims to address such a gap by providing a fully integrated system with both software and hardware capabilities that can automatically partition an object into smaller chunks to be assigned to individual robots and generate a schedule for collision-free printing.

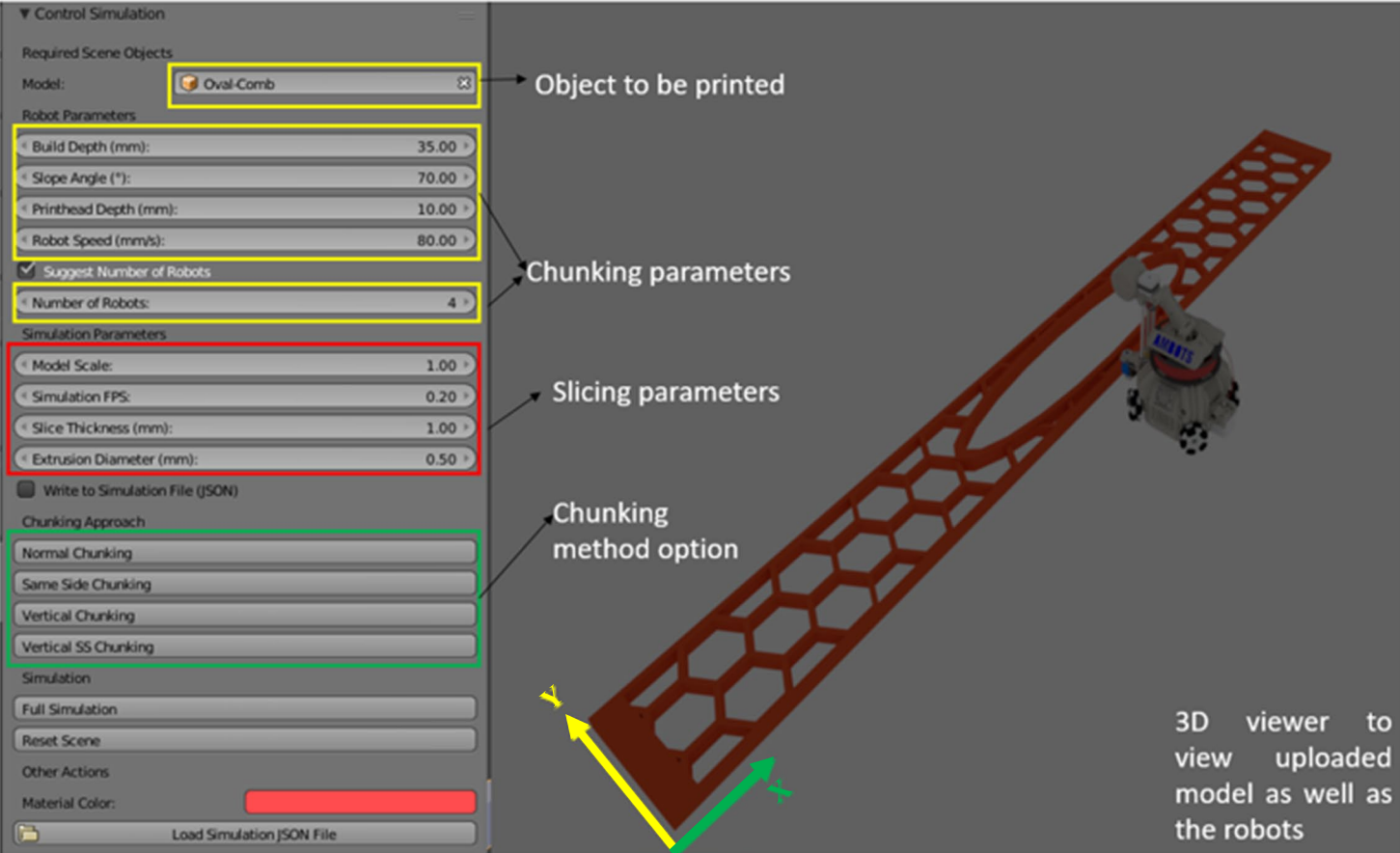


Figure 3: The UI of a custom addon to Blender that allows user to choose different chunking parameters as well as the slicing parameters for C3DP.

### 3. ARCHITECTURE OF THE C3DP SIMULATION

To realize C3DP, an integrated software system is required to streamline the process of chunking, scheduling, slicing, and simulating before the printers are committed to printing. However, there is no existing system for multi-robot C3DP. To address this gap, we developed a system with all these functionalities integrated. Its architecture is presented in **Figure 2**. Once a user uploads a CAD model of the part to be printed, it is first chunked by the *Chunker* using the chunking parameters specified by the user (e.g., the slope-angle, the width of chunks, and the depth of chunks). Then, the *Scheduler* generates a print schedule of how the chunks will be printed in sequence and in parallel by considering chunk dependencies as well as the available resources, such as the number of robots. Chunk assignment is done along with the print schedule in this step. After that, the *Slicer* generates G-code for each of these chunks, and the *Simulator* uses the generated G-code to animate the print schedule and process. In the following sections, we describe the methods and algorithms that enable each of these individual modules.

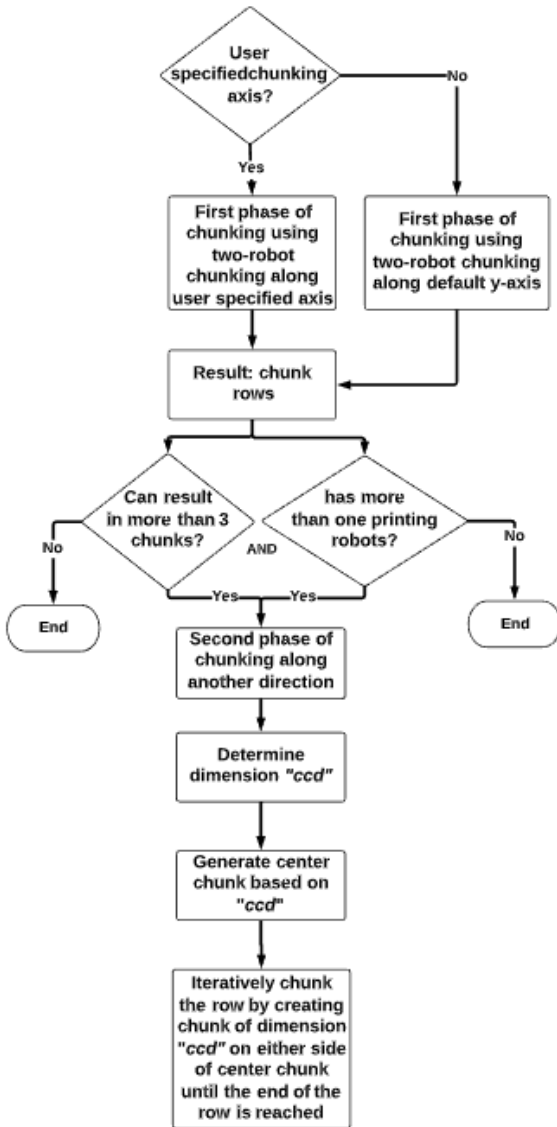
#### 2.1 Chunker

The *Chunker* takes the CAD model of a part as an input and splits the CAD model into smaller chunks based on the parameters set by the user. The chunking needs to be done in such a way that it facilitates simultaneous multi-robot printing. Thus, we use the sloped-surface chunking strategy [25] to support the cooperation. While we use chunk-based cooperation between the robots (i.e., divide a part into chunks to be assigned to robots so that the cooperation between the robots takes place at the chunk level), such cooperation can happen at different scales. For example, Project Escher used layer-level cooperation between multiple printheads, where each printhead is responsible for printing its own portion, which is not overlapping with other portions at each individual layer. Similarly, concurrent printing [18] used G-code layer cooperation between the robots where each G-code line is assigned to different printheads and checked for what lines can be printed concurrently without collision given the available number of printheads. However, further studies of the cooperation at the different levels are needed to ensure the mechanical properties are comparable to those of the traditional printed part.

Once the CAD model is uploaded to the system, a user interface (UI), as shown in **Figure 3**, is available for users to choose different options for chunking parameters. The yellow rectangular bar in **Figure 3** highlights different parameters that the user can set. This includes: 1) build depth (i.e., the depth of chunk in the y-direction), 2) slope angle of chunking, 3) printhead depth of the robot (i.e., how far the robot's print nozzle can reach), 4) the number of available robots. If a user does not want to specify the number of robots, the system can suggest the optimal number of robots based on the dimensions of the part and the robot. The user can also pick a different chunking strategy to chunk the part. More detailed information regarding the selection of chunking parameters is presented in our previous study [26]. In this paper, an enhanced version of the *Chunker* that includes chunking capability for multiple robots is presented. Particularly, we present two additional chunking options: the divide-and-conquer chunking and the same-side chunking.

Each of the chunking options has its benefits for a specific type of situation. Below are some of the guidelines for choosing a proper chunking option based on resource availability, space availability, and the dimension of the part to be printed.

1. If the user only has two robots available for printing, the previously developed two-robot sloped-surface chunking strategy will be chosen as default by the system.
2. If the user has more than two printing robots and the printing space is unlimited, then the divide-and-conquer strategy would provide the best outcome.
3. If the user has a limited printing space, where the printer can only be placed on one side of the part, the same-side chunking strategy will be called.



**Figure 4:** Flowchart demonstrating the divide and

For example, if a part is located at a location where the robot can only move in one direction and no room beyond that location is available, the same-side chunking must be used.

In the following sections, we describe each of these chunking options in a more detailed manner.

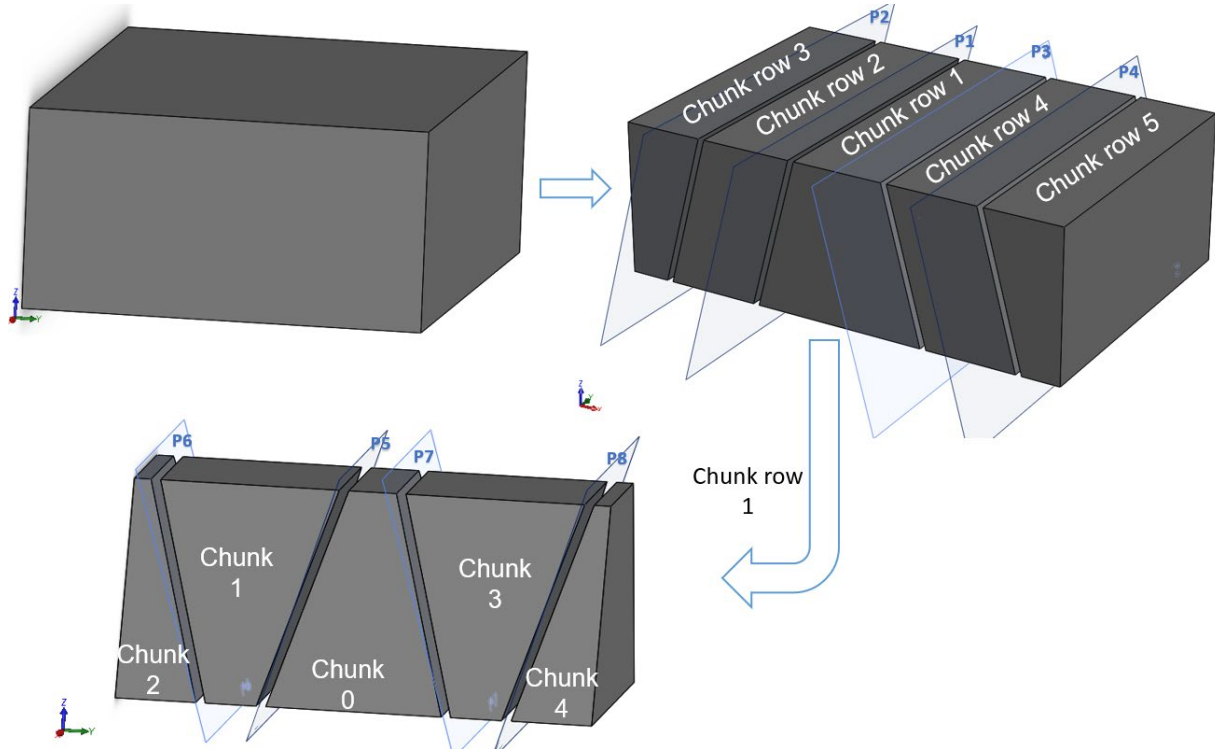
### 3.1.1 Divide-and-Conquer Chunking

This chunking option extends the two-robot chunking to multi-robot chunking, where the *Chunker* first centers the object and then splits the object into multiple chunks along one direction with the sloped surface using a bisecting algorithm, as shown in **Figure 4**. In this strategy, the first phase of chunking takes place along the axis (either along the x-axis or y-axis) specified by the user. Allowing users to specify an axis gives them a choice on whether to chunk along the length of the part or width of the part first. If the user does not specify the axis, the chunking is done along the default y-axis. If the second phase of chunking is required, i.e., if chunking is required in both axis, which depends on the number of available robots and the dimensions of the resulting chunk row, it takes place along the other axis that is different from the axis used in the first phase. The center chunk dimension ( $ccd$ ) depends on the reach of the robot (how far the robot can print from its current location), and the slope angle chosen by the user. The dimension is calculated using the slope angle of the chunk and the reach of the printing robot as presented in **Equation (1)**. Chunking planes are created on each side of the centerline such that they are separated from the centerline by the distance of  $ccd$ . The chunks are created by bisecting the part at these chunking planes. This process is iterative and takes place on both sides until the entire part is chunked, as shown in **Figure 5**.

$$ccd = R + \tan \theta \times z_{height} \quad (1)$$

where  $R$  is the reach of the robot,  $\theta$  is the slope angle and  $z_{height}$  is the height of the part.

The resulting number of chunks depends on the number of available robots, the desired size of the chunk, and the size of the part itself. The width of the chunk must be large enough so that the two adjacent robots would not collide during the printing of chunks.

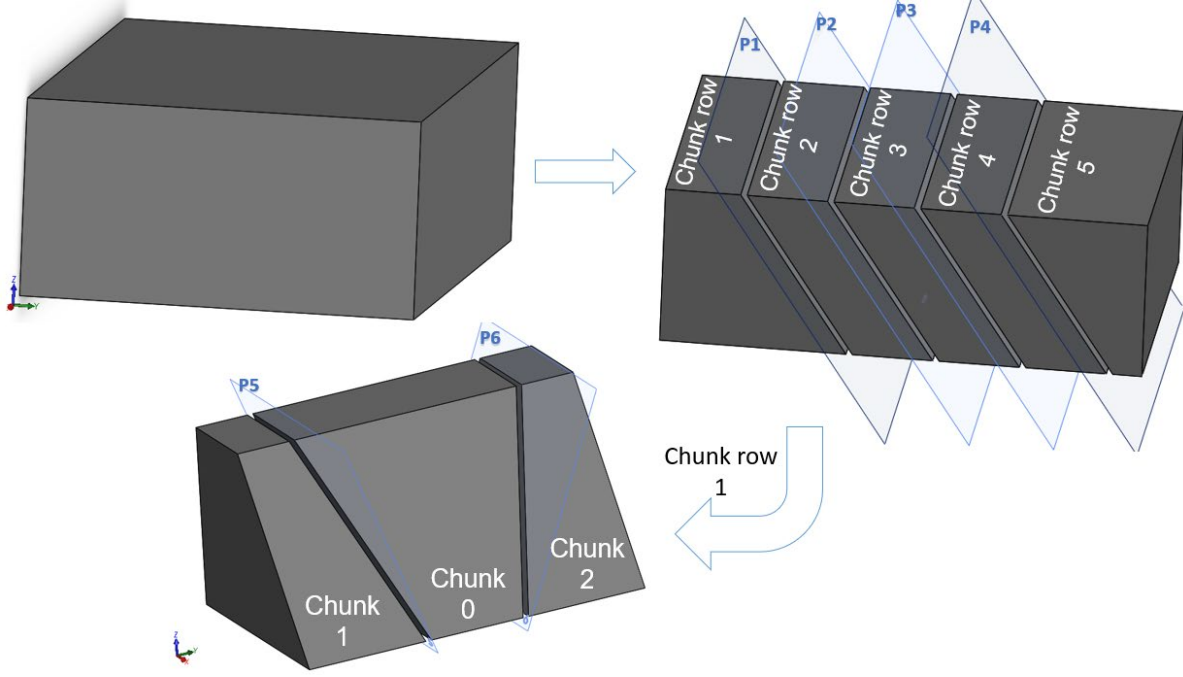


**Figure 5:** Chunking process of a rectangular bar. First, the chunking takes place along direction (x-direction). Center chunk is created first and moves out in both direction in iterative manner to split the part further until the end of the part. Second phase chunking, shown in bottom, (only center shown for clarity). First, a center chunk is created. The row is then iteratively chunked along the longitudinal direction until the end of the chunk row is reached.

Thus, the width of the chunk is

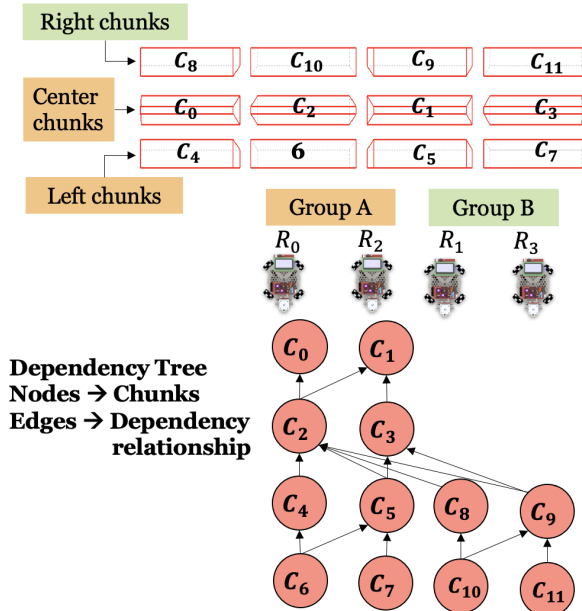
$$Chunk\_width = \max(W_r, \frac{w_p}{num\_robots}) \quad (2)$$





**Figure 6:** Demonstration of how same-side chunking takes place in Chunker (The space between the chunks is shown for demonstration purpose only. No such space exists in reality). The second phase of chunking demonstrated using only single row for clarification

where,  $W_r$  is the width of the printing robot (it is equal to the reach of the robot for SCARA printers, whereas, for mobile printers (similar to our first generation of robots [27]), it is not),  $W_p$  is the width of the part along a specified chunk direction and,  $num\_robots$  is the number of available robots for printing. Once the width of the chunk and the number of chunks are determined, the chunking plane is created by iteratively spacing  $chunk\_width$  from the centerline of the chunk row. The angle of the plane is alternated at every chunking plane as shown in **Figure 5**, i.e., if a chunking plane is created at  $45^\circ$  first, the next chunking plane is created at  $135^\circ$  and so forth. Doing so allows us to create chunks with alternating slopes, allowing multiple robots to print simultaneously in each row.



**Figure 7** Chunk Assignment of the robots in scaled-chunker method

### 3.1.2 Same-side chunking

The same-side chunking applies to the scenarios where space only allows the printing robots to be on one side of the center row chunk. Thus, instead of starting from the center and moving outward in both directions (as in the divide-and-conquer strategy), same-side chunking allows printing to start from one end and extend in only one direction. The same-side chunking uses a similar chunking method as described in the divide-and-conquer approach. The only difference lies in the location of the origin chunk. The origin chunk is created at the center of the part in the divide-and-conquer chunking, whereas it is created at one of the ends of the part in the same-side chunking. Once the origin chunk is created, the chunking plane is iteratively shifted in one direction rather than two directions. The second phase of chunking takes place in the same manner as the divide-and-conquer approach. **Figure 6** presents the origin chunk in the same-side chunking and the iteration of the chunking plane along one direction until the end of the part is reached.

### 3.2 Scheduler

Once the chunks are generated, they will be assigned to individual robots and scheduled for printing based on their dependency relations. The dependency relation is generated since adjacent chunks share the same sloped surfaces, and the chunks with overhanging slopes cannot be printed

before their adjacent chunks that support the overhanging slopes. As shown in **Figure 6**, Chunk0 between plane P5 and P6 cannot be printed prior to printing Chunk1 and Chunk2 because of the overhang.

The details of how the chunk assignment takes place and how the dependent relationship is used for scheduling is presented in our previous study [28]. This section discusses how the scheduling strategy is encoded into the software to achieve collision-free printing. For two-robot printing, a simple strategy is adopted where the origin chunk is assigned along with all the chunks on the left side to one robot and the remaining chunks on the right side to the second robot. As the chunking becomes more complicated, the complexity of the chunk assignment increases. The approach for the chunk assignment is presented in **Figure 7**. For simplification, all the robots are divided into two groups randomly, then the assignment approach used in the two-robot printing scenario is implemented. After that, the chunks in each group are assigned among the robots ( $R_1, R_2, R_3, R_4, \dots$ ) in their respective groups. If  $C_c$  represents center row chunks (for example, the chunks 0, 1, 2, and 3 in **Figure 7**),  $C_L$  represents left row chunks (for example, the chunks 4, 5, 6, and 7 in **Figure 7**), and  $C_R$  right row chunks (for example, the chunks 8, 9, 10, 11 in **Figure 7**) and the total available robots are randomly divided into two groups,  $G_A$  and  $G_B$ . Then, all the chunks in  $C_c$  and  $C_L$  are assigned to  $G_A$  and the rest to  $G_B$ . The assignment of chunks to the group is done based on human heuristics and might not be optimal.

$$\begin{aligned}
C_c &= [C_{c_1}, C_{c_2}, C_{c_3}, \dots, C_{c_n}] \\
C_L &= [C_{L_1}, C_{L_2}, C_{L_3}, \dots, C_{L_n}] \\
C_R &= [C_{R_1}, C_{R_2}, C_{R_3}, \dots, C_{R_n}] \\
G_A &= [R_1, R_3, \dots, R_{m-1}] \\
G_B &= [R_0, R_2, \dots, R_m] \\
G_A &\rightarrow \{C_c, C_L\} \\
G_B &\rightarrow \{C_R\}
\end{aligned}$$

Once the chunk assignment is completed, a print schedule is generated based on the chunk dependency following the rules below:

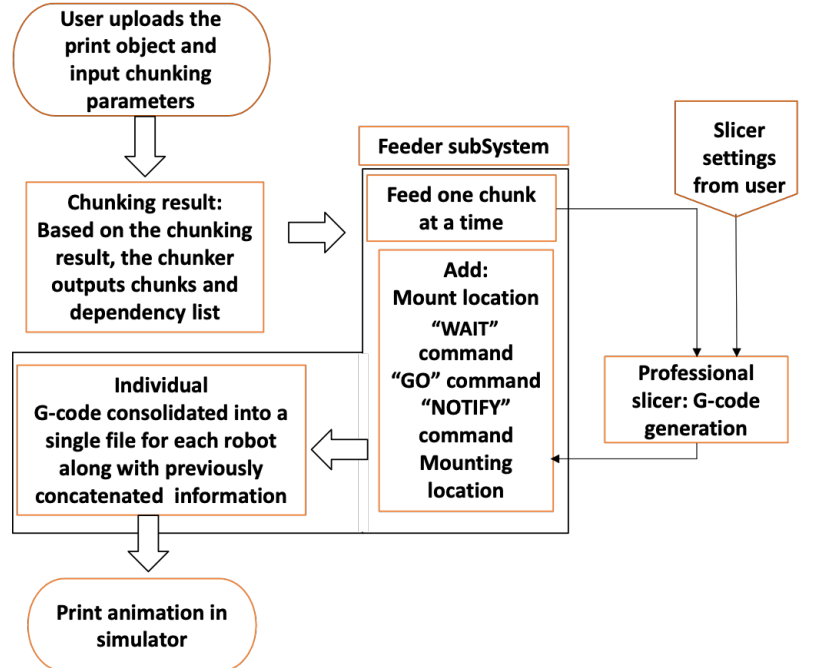
- The chunks with no dependencies are printed first.
- Once those chunks are printed, the chunks with already-printed chunks as dependencies are chosen for printing. This process iterates till all the chunks are printed.

More detailed information on how to carry out a schedule based on a dependency tree is presented in our previous study [28].

### 3.3 Slicer

In our previous study [26], we developed a simple custom-designed *Slicer* that can generate toolpaths for all the printing robots based on the chunks assigned. To handle more complicated geometries with higher efficiency and robustness, we used an open-source slicing engine, Cura, and developed a software module to support the cooperation of multiple robots. Fundamentally, for two robots to cooperate, they must be able to align in space and time. The spatial alignment is enabled by the positioning mechanisms embedded in our hardware platform, as described in Section 4. The temporal alignment is realized with a pair of custom G-code command, “WAIT” and “NOTIFY”, which are used by robots to tell each other via wireless communication when they should pause and when they may proceed to execute the next line of G-code. A “WAIT” command is inserted at the beginning of any chunk that has chunk dependencies, i.e., another chunk has to be printed before printing the current chunk. A “NOTIFY” command is added at the end of the G-code of the chunk on which another chunk depends. This notifies another robot that the dependency has been satisfied, and it can go ahead with printing the chunk.

To utilize a third-party slicing engine for C3DP, we developed a *Feeder* system. The *Feeder* system imports individual STL models of the chunks output by *Chunker* one at a time to generate the G-code of each chunk using the professional *Slicer* based on the settings, such as infill density, print speed, layer height, etc. Once the G-code files for all the chunks assigned to an individual robot are generated, additional transitional information, such as the printing sequence, “WAIT” and “NOTIFY” commands, is added between the G-code files of chunks based on the generated schedule. In the end, all the G-code files are combined into one file and sent to the robot. The entire process is demonstrated in **Figure 8**.



**Figure 8.** Once the part is uploaded and chunked, the feeder subsystem feeds one chunk at a time and send it to slicer. Once the G-code is created, the other relevant information is added to the G-code. The G-code of all chunks of a robot are consolidated into one single file and used for visualization

### 3.4 Simulator

The *Simulator* animates the printing process and provides a visualization of how the printing unfolds and how the final product looks like. In our system, the *Simulator* is developed in Blender environment. To generate visualization, frames are generated at a specified rate of frames per second (30 fps is used in this study as a default), and the user can choose to either increase or decrease fps based on their desired quality of outcome. A more detailed explanation of how the simulator works can be found in our prior work [26].

## 4. HARDWARE PLATFORM DEVELOPMENT

The hardware platform has gone through major updates since the first generation [27]. This current hardware platform consists of four main components:

1. A selective compliance assembly robot arm (SCARA) 3D printer.
2. A mobile platform that transports the SCARA printer from one location to another after the completion of a chunk printing.
3. A modularized floor tile system that assists the navigation of the mobile platform allows SCARA printers to be mounted on and provides a power supply and a charging station for the mobile platform.
4. A wireless network that coordinates the communication between SCARA printers and mobile platforms.

A prototype of the current design of the hardware platform is presented in **Figure 9**. In a typical printing process, the assembled G-code files from the *Feeder* system will be sent to their assigned SCARA printers. To begin printing:

- The mobile platform reaches the current location of the SCARA printer, and it notifies the printer to mount onto the platform and unmount from the floor tile.
- Once mounting and unmounting is complete, a custom G-code command “MOVE” then calls the mobile platform to transport the SCARA printer and maneuver it to its destination. During transportation, while the printer is mounted on the mobile platform, it is powered using the battery pack installed on the mobile platform.
- After reaching the destination, the platform notifies the printer to start mounting into the floor and unmounting from the mobile platform.
- The SCARA printer will then start to execute the rest of the G-code until it hits a “WAIT” command. The “WAIT” command notifies the printer that, in order to print the current chunk, its dependent chunks have to be printed first. Once the dependent chunks are printed by other SCARA printers, another custom G-code command, “NOTIFY” will be sent to the printer that has been waiting to start printing the next chunk. This MOVE-WAIT-NOTIFY process continues based on the generated manufacturing schedule till all the chunks are printed. Currently, only one SCARA printer can request the mobile platform at a time. If multiple printers request transportation, the first-in-first-out rule is used to break the tie.

**Table 1.** Technical parameters of the new 3D printing SCARA printer

XY reach	50mm-350mm
Max Z-height	300 mm
Filament Feed	Bowden, 1.75mm
Nozzle	Single extruder
Maximum Temperature	295° C
Hot End	Single extruder
X/Y Motion	2 axis SCARA
Z motion	300 mm guide motion driven by lead screw
Layer resolution	10 $\mu$ m
Print Speed	50 mm/s
Print repeatability	5 $\mu$ m
Power Input	build floor, battery pack via mobile platform
Power consumption	20 W
Software	compatible with open-source software
Connectivity	Wireless

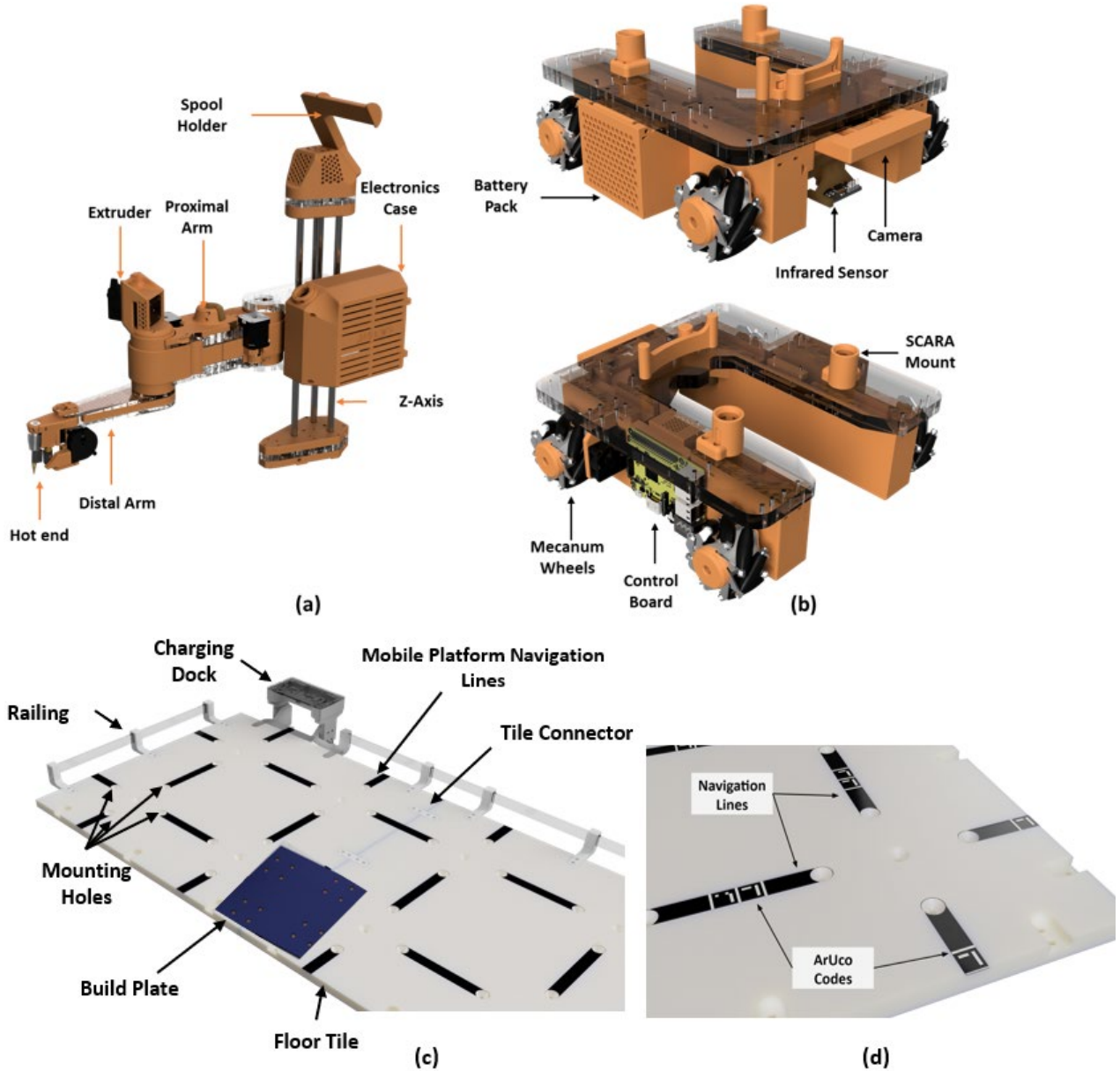
### 4.1 SCARA Printer

With the wide adoption of robotic arms in modern factories, we developed a SCARA 3D printer for filament extrusion, as shown in **Figure 9(a)**. This design would facilitate an easier transition to future factories. The main functionalities of the SCARA printer are:

1. *Print the assigned chunks:* The SCARA printer is equipped with a single extruder and uses a 1.75 mm filament to print assigned chunks. The print specifications are presented in **Table 1**. The printer is equipped with three mounting leads that screw into the nuts installed in the mounting holes in the floor tile. While the printer is printing, it is mounted using those lead screws to prevent any movement. These lead screws also work as a connector to supply power to the printer from the power source installed in the floor tile. Such a locking mechanism allows the printer to minimize vibration as well as tipping over, which could be a concern, especially when the arm is fully extended.



2. *Communication with the mobile platform:* The SCARA printer is immobile during printing but communicates with the mobile platform wirelessly if it needs to be moved from one location to another. The printers have two additional mounting pegs to mount onto the mobile platform for secure transportation. These mounts also act as connectors to supply power from the battery



**Figure 9.** (a) The updated 3D printer, equipped with scara-arm (b) The new mobile platform that transports scara arm from one print location to another (c) Modular floor tiles (d) Navigation lines and ArUco codes in floor tiles for navigation

pack of the mobile platform to the printer during transportation. This, along with the power source installed in the floor tile source, ensures that the SCARA printers always have at least one source of power at any time.

The maximum reach of the SCARA is much larger than the previous generation of the robot. It can reach from 50 mm to 350 mm, i.e., it cannot print anything that is closer than 50 mm to its mounting location and can reach 350 mm when it is fully extended. Currently, the auto-calibration system uses an auto bed leveling sensor and a four-point calibration system, one for each corner of the build plate.

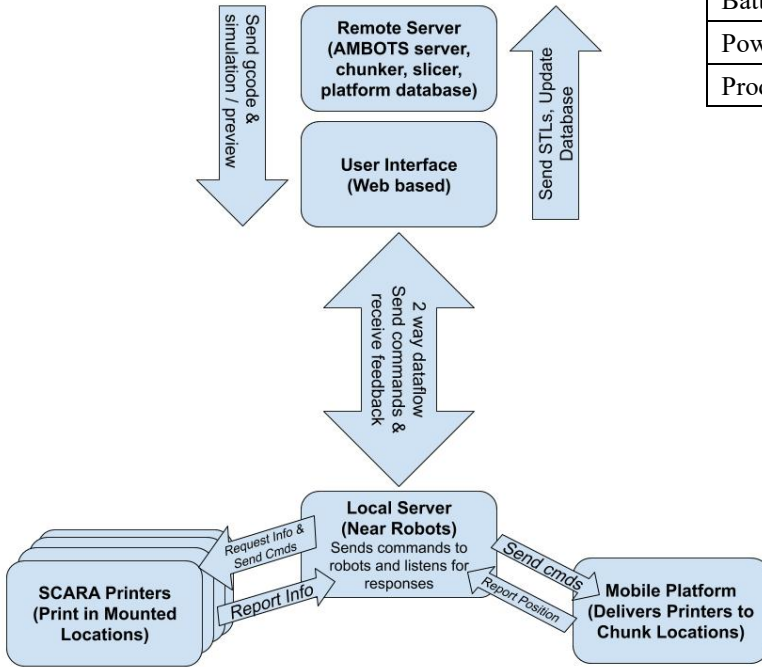
## 4.2 Mobile Platform

Because the printing robots spend most of the time printing and little time moving between chunks, we separated the mobile platform from the SCARA printer. This new design (see **Figure 9(b)** and **Table 2**), on the one hand, enhances the stability of the SCARA printers during printing, and on the other hand, reduces the overall cost by allowing multiple SCARA printers to share one mobile platform. The main functionalities of the mobile platform are:

1. *Navigate on the floor*: The mobile platform is equipped with mecanum wheels due to its omnidirectional property for maneuverability. In order to maneuver from one location to another accurately, the platform is also equipped with infrared sensors. These sensors scan the barcode on the floor tile to get accurate positional information and add a layer of position verification. The use of a barcode ensures that the platform moves to the location where it is supposed to be. Thus, this positional information is used in conjunction with the positional movement that is tracked by the encoder of the motors to rectify any positional discrepancies that might occur due to unforeseen mishaps. In addition to the sensors, it is also equipped with a camera that can utilize computer vision to verify positional accuracy.
2. *Carry the SCARA printer and power supply*: The mobile platform is responsible for transporting the SCARA printer from one location to another safely. To achieve this, it is equipped with two mounting pegs with holes in the center on the top. Once the SCARA printer's location is reached, the printer is notified wirelessly, and the printer starts lowering the leads to mounting into the holes of the platform. These holes also act as a female connector that connects with the male connector of the printer to supply power so that the printer can mount and unmount

**Table 2.** Specifications of the mobile platform

Wheels	Mecanum wheels, omnidirectional
Travel Speed	60 mm/s
Navigation	Infrared sensors, camera
Communication	Wireless communication
Power Source	Battery pack, charging station
Battery life	2 hours use, 6 hours idle
Power Consumption	20 Watts
Product dimension	25 cm×33 cm×9 cm



**Figure 10.** The information flow in the cooperative 3D printing system between different components

while it is not in connection with the floor tile. Once the printer finishes printing in the current location, it unmounts from the floor tile and is transported to its next workstation.

3. *Autonomous charging*: The mobile platform is equipped with a rechargeable battery pack that has a life cycle of 2 hours of continuous use and an idle time of 6 hours. But if it is not being used for more than 5 minutes, it automatically retreats to a charging station located in one of the corners on the build floor.

## 4.3 Floor Tile

A modularized floor tile system is developed to provide a base for the C3DP platform. It consists of modularized floor tiles and connectors between tiles for scaling to a size as large as the factory size. Each floor tile is 600mm × 600mm with four mounting locations spaced at 300mm apart in both X and Y directions, as illustrated in **Figure 9(c)**. The modular floor tiles have a slot on each side which is used to align two-floor tiles

together. A connector that sits flush in this slot can be used to attach floor tiles with high accuracy. The floor tile system provides the following functionalities:

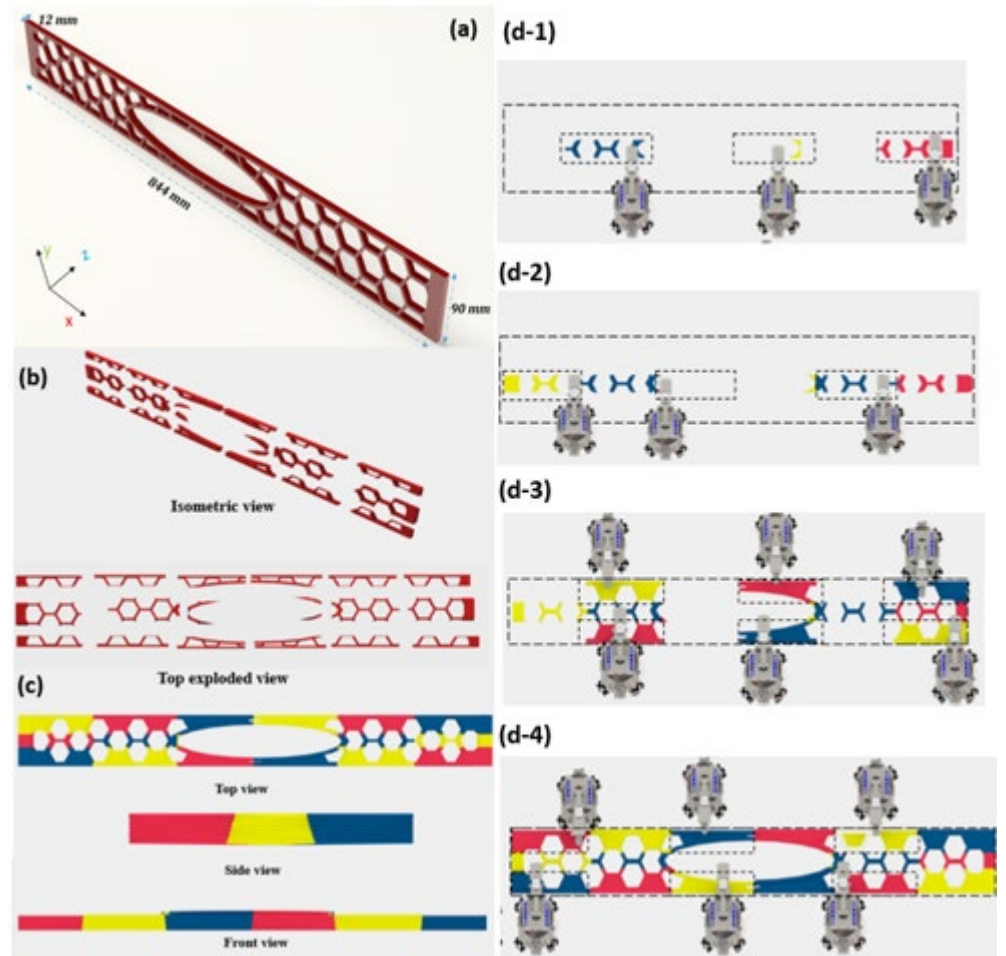
1. *Mounting and powering the SCARA 3D printers*: Each of the four mounting locations consists of five mounting holes. Nuts are installed inside these mounting holes so that the lead screws in the SCARA arm can mount into them. In order to auto-correct the slight misalignment due to tolerances stacking, each mounting hole has a countersink on the top surface. In addition, the floor tile system is equipped with electrified stainless-steel conductive strips of alternating polarity on the bottom side of

the floor tile. Once the leads of the SCARA printer mount into the mounting holes, they come in contact with the conductive strips, which transfers power to the SCARA arm. A rectifier circuit is used to sort the polarity of the current to ensure that current flows are in the correct direction regardless of the polarity of active strips. This powers the SCARA printer during the entirety of the printing and realizes wireless charging.

2. *Assist navigation of mobile platform:* The floor tile is equipped with navigation lines of black colored markers to contrast with the floor tile. These navigation lines are followed by infrared sensors to maneuver from one mount location to another. In addition to having navigation guides, the floor tile also includes an embedded barcode system to provide positional information. This barcode includes positional information such as to what row and column it belongs to (**Figure 9(d)**). This is used by the camera installed on the mobile platform to get positional awareness.
3. *Provide a charging station for the mobile platform:* The floor tile is also equipped with a charging station mounted onto the edge of the tile. This charging station resembles a cart storage area outside a supermarket, where a mobile platform can park itself into. Once the platform is aligned, a connection is formed between the charging station and the platform, initiating the charging of the platform.
4. *Provide a base to mount build plate:* Build plate is a  $300\text{mm} \times 300\text{mm}$  modular square block where a part is printed on. It is mounted on the build floor with a base that has four conical pegs resembling countersink tools. Having such pegs helps align the build plates at the correct location and prevents the movement of the build plate in the XY plane.

#### 4.4 Wireless Network

The communication between the different components of the hardware takes place via a wireless communication network. Both the SCARA printer and the mobile platform are equipped with a wireless communicator connected to a local server via the wireless network. The local server collects all the pertinent information required for planning, coordination, and cooperation among different printers. This local server takes information from the software components, such as G-code information for each SCARA printer, then transmits it to the individual SCARA printers wirelessly. This flow of information between the different components of C3DP is presented in **Figure 10**.



**Figure 11.** (a) A honeycomb bar with dimensions (b) Different view of chunked bar (c) Different view of sliced honeycomb bar (d) Snapshots of print simulations at different timesteps

## 5. VALIDATION OF THE SYSTEM

### 5.1 Simulation

To validate and demonstrate the capability of the developed C3DP system, we have conducted two case studies. The first case study demonstrates the system’s scalability, and the print object has a relatively simpler geometry. The second case study aims to demonstrate the capability of the software in handling complicated geometries.

#### Case I: A bar with a honeycomb design

The first print object is a rectangular bar with an internal honeycomb structure. Its dimension is  $844mm \times 90mm \times 12mm$  (see **Figure 11**). The chunking parameters used are presented in **Table 3**.

In total, 18 chunks are obtained from the *Chunker*, and the chunks are shown in **Figure 11 (b)**. The *Chunker* also outputs the chunk dependency in JSON format, which is then used to generate a print schedule with chunk assignment. Both the print schedule and the chunk assignment are exported in JSON file by the *Scheduler*. The *Slicer* then generates G-code for every chunk and adds other pertinent information related to the position and dependency. A single G-code file is created for all the chunks assigned to the same robot. This G-code file contains printing commands as well as the “WAIT” and “NOTIFY” commands to implement dependency between chunks. In addition to this, it also contains the command to notify the mobile platform of any robot’s transportation requirement. **Figure 11 (c)** shows the different views of the part obtained after slicing. The different color of the chunks does not represent the actual color of the chunks to be printed but is used to differentiate a chunk from adjacent chunks. The entire process (chunking, slicing, and scheduling) for this case study took roughly about 10 seconds and normally could range between 5-10 seconds. However, the majority of the time can be attributed to the slicing operation.

Once the G-code files for all the robots are created, the printing is simulated. The *Simulator* visualizes the print sequence and the entire printing process. The screenshots of the simulation at different time steps are presented in **Figure 11(d-1 through d-4)**. First, the chunks that do not have any dependencies are printed (e.g., the three chunks in the center row as shown in **Figure 11(d-1)**). Once those are completed, the same robots move over to print the gap chunks, as shown in **Figure 11(d-2)**. After finishing the center row, half the robot moves to one side of the row, and the other half moves to another half and starts working on the chunks as shown in **Figure 11(d-3)** and **11(d-4)** until the part is completed.

#### Case II: A Razorback Cutout

The second object is a Razorback cutout shown in **Figure 12(a)**. It has a dimension of  $610mm \times 267mm \times 52mm$ . The chunking and slicing parameters used in this case study are listed in **Table 4**. The output of the *Chunker* for the Razorback cutout is presented in **Figure 12**. The chunking parameters resulted in 24 chunks. Once the chunking is complete, the schedule will be generated based on the chunks’ dependencies. The *Slicer* generates G-code for different chunks to be printed by individual robots, one at a time, and outputs the G-code files. The output of *Slicer* is presented in **Figure 12 (c)**, and the snapshot of the simulation is presented in **Figure 12(d)**. Since the print object is large in two directions, the entire process took slightly longer than it did for the first case study. The overall process took less than 15 seconds and typically could range between 10-15 seconds. That being said, the slicing operation is the most cumbersome in terms of computation time. In addition to this, it needs to be clarified that the computation time could change based on the chosen parameters, especially the parameters for slicing operation. For example, the aforementioned computation time is valid for slicing an object with 0.40mm layer height. Slicing the same object with 0.15 mm layer height could take a larger computation time (~1 minute).

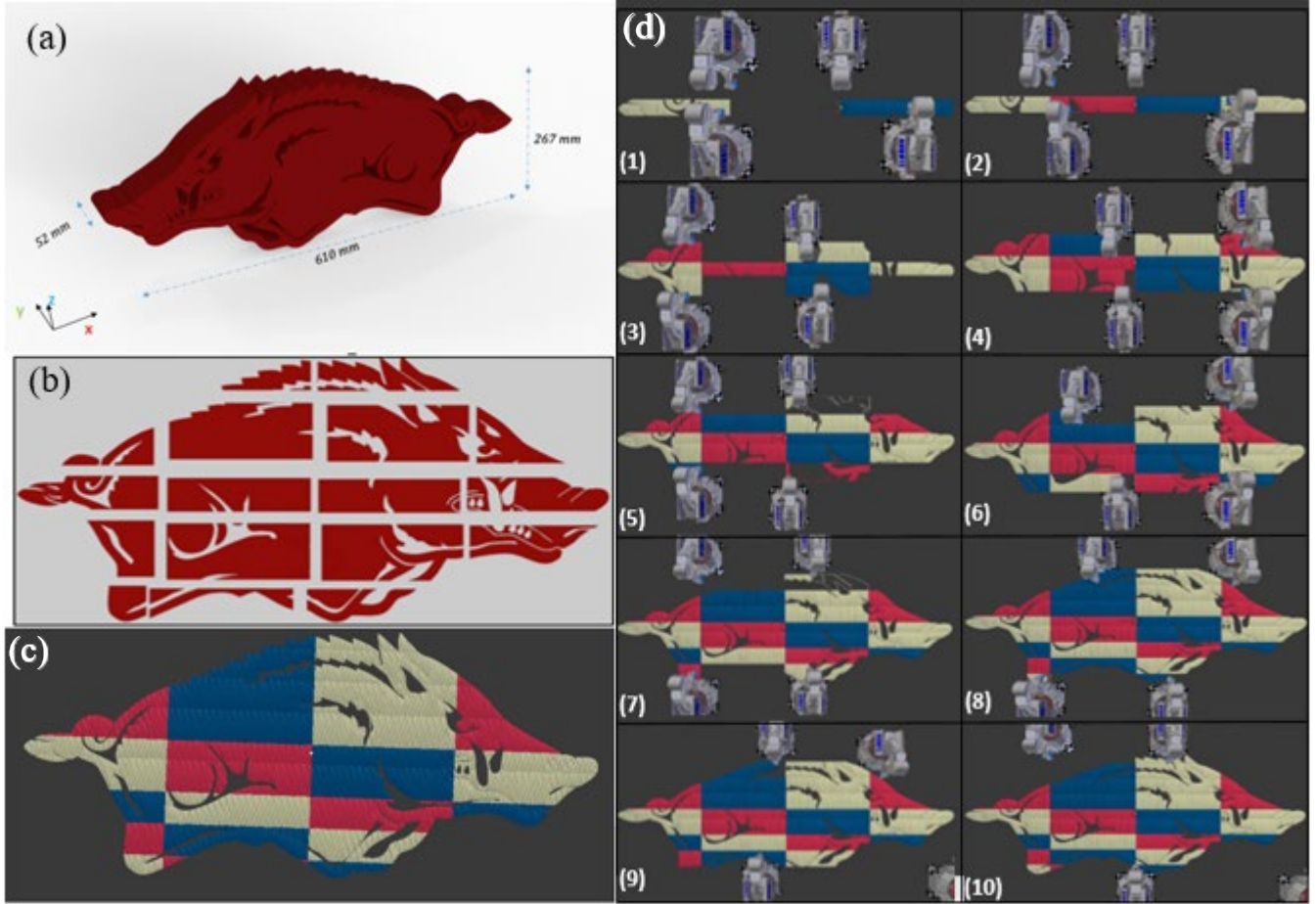
**Table 3** Parameters used for first chunking and then slicing the rectangular bar

Parameters	Value (units)
Number of robots	6
Slope Angle	70 degrees
Build depth	35 mm
Printhead depth	10 mm
Chunking Option	Divide and conquer

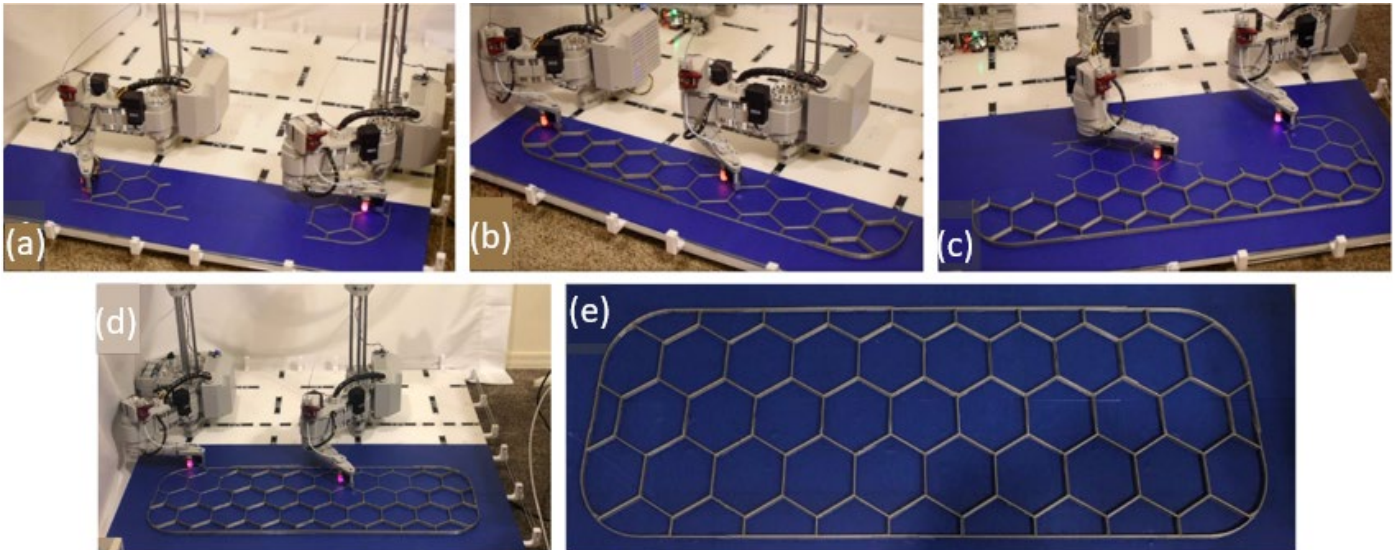
**Table 4** Parameters used for first chunking and then slicing the Razorback cutout

Parameters	Value (units)
Number of robots	64
Slope Angle	70 degrees
Build depth	70 mm
Printhead depth	20 mm
Chunking Option	Divide and conquer





**Figure 12.** (a) The Razorback cutout with specified dimension (b) Razorback chunked into 24 chunks (c) Chunked Razorback after slicing (d) C3DP print simulation using 6 robots



**Figure 13.** Snapshots of actual printing using two SCARA printers

## 5.2 Physical Validation





**Figure 14.** (a) AMBOTS logo printed using two SCARA printers (b) Result after post processing

To test the overall integration of the system, we conducted two large print tests and physically validated the system. The first print test includes a large honeycomb artwork of dimension  $1000\text{ mm} \times 350\text{ mm} \times 10\text{ mm}$ . The part extends over eight build plates, and the printing requires two SCARA printers, one mobile platform, sixteen floor tiles, and the software components. The print test consisted of eight chunks and took a total of 24 hours to complete using two SCARA printers. The snapshots of the print test are presented in **Figure 13**, and the demo video is published online<sup>1</sup>. The second print test includes the AMBOTS logo. The logo has a dimension of  $1200\text{ mm} \times 250\text{ mm} \times 75\text{ mm}$  and took roughly 72 hours to complete printing using two SCARA printers. The chunking resulted in three chunks, which were assigned to the two printers. The final completed print is presented in **Figure 14(a)**, and the one after post-processing is presented in **Figure 14(b)**.

## 6. CONCLUSION AND FUTURE WORK

While different manufacturing paradigms have been proposed over the years, these paradigms focus on developing complex global networks with intricate supply chains to meet the changing demand of customers. They overlook a simple solution where a factory can be established in a local ecosystem capable of producing multiple different products. Swarm manufacturing does exactly that, where teams of mobile robots work together on a factory floor to manufacture desired parts based on customer demand. This paradigm of swarm manufacturing focuses on shifting from traditional manufacturing based on conveyor belts to multi-robot cooperative manufacturing where multiple different products can be manufactured on the same factory floor without costly tools change. Rather than focusing on improving the efficiency of mass production, swarm manufacturing focuses on new customer demand for mass customization. In this paper, we presented an integrated C3DP architecture with all the necessary software and hardware components that can enable swarm manufacturing, one of the implementations of swarm manufacturing. While the architecture is demonstrated for multi-robot 3D printing, it is general enough for any other manufacturing method as it focuses on solving the issue of cooperation of multiple robots in shared space while manufacturing.

The software component includes five critical subsystems. The *Chunker* divides a part into small chunks with different options, such as the divide-and-conquer method and the same-side chunking. Users can choose the most appropriate strategy based on parts' geometries and their preferences. The *Scheduler* takes the dependencies between the chunks generated during the chunking process to generate print schedules. The *Scheduler* also automatically assigns chunks based on the number of available robots. The current C3DP system adopted a more robust slicer called Cura, and we developed a *Feeder* system to adapt it to the C3DP environment. The same feeder adds pertinent information that includes but is not limited to mounting location for the printer, the "WAIT" and "GO" commands to enforce the print schedule, etc. The *Simulator* generates an animation based on the generated G-code.

The hardware includes a new SCARA printer and a separate mobile platform for printer transportation. In addition, a new modular floor tile is developed that serves as the power source for the SCARA printer and provides positional information to the mobile platform. Finally, in order to validate the system, two case studies are conducted: a rectangular bar and a Razorback cutout, where all the five components of the software system are tested and verified.

The major contributions of this study are in three aspects:

1. This is the first time a fully integrated and functional C3DP platform is presented, taking the first step towards swarm manufacturing using multiple mobile robots for manufacturing.
2. We developed new chunking strategies that can be used to subdivide a part into smaller chunks and utilize multiple mobile robots to print them cooperatively.

<sup>1</sup> <https://www.youtube.com/watch?v=xMMauLmHcwk>

3. We successfully adapted a conventional slicer into the C3DP environment.

This is the first time a fully integrated and functional C3DP platform is presented, taking the first step towards swarm manufacturing using multiple mobile robots for manufacturing. While promising, there are a few limitations that need to be addressed in the future. First, the use of the SCARA arm poses many constraints on the spatial movements of the toolheads and therefore limits the chunking options and conflict-free scheduling. Using a robotic arm with more degrees of freedom will enable more freeform chunking and more flexible scheduling options. Second, the maximum height of the SCARA printing robot is currently 300 mm. To print taller objects, taller robots will be needed. In addition, the placement of the build plate and replacement of the filaments are currently manual, and new subsystems are needed to automate these tasks. Moreover, to achieve the full potential of the platform, an AI-powered software is needed to enable the robots to sense the environment and make autonomous decisions to finish tasks in the queue in a decentralized and autonomous fashion.

## 6. ACKNOWLEDGEMENT

This project is supported by the National Science Foundation (NSF) Division of IIP through Grant # 1914249 and the commercialization fund through The Office of Vice Chancellor for Economic Development from the University of Arkansas.

## REFERENCES

- [1] N. R. Council, "Visionary manufacturing challenges for 2020," *CHEMTECH*, vol. 29, no. 5, pp. 49–56, 1999.
- [2] K. Ishii, C. Juengel, and C. F. Eubanks, "Design for product variety: key to product line structuring," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 1995, vol. 17179, pp. 499–506.
- [3] R. B. Stake, "A hierarchical classification of the reasons for dividing products into modules: a theoretical analysis of module drivers." Institutionen för produktionssystem, 1999.
- [4] A. De Toni and S. Tonchia, "Manufacturing flexibility: a literature review," *Int. J. Prod. Res.*, vol. 36, no. 6, pp. 1587–1617, 1998, doi: 10.1080/002075498193183.
- [5] H. A. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *Int. J. Flex. Manuf. Syst.*, vol. 17, no. 3rd International Conference on Reconfigurable Manufacturing, pp. 261–276, 2005, doi: 10.1007/s10696-006-9028-7.
- [6] Y. Koren *et al.*, "Reconfigurable manufacturing systems," *CIRP ANNALS 1999: MANUFACTURING TECHNOLOGY, VOL 48 NO 2 1999*, no. 49th General Assembly of CIRP. pp. 527–540, 1999. doi: 10.1016/S0007-8506(07)63232-6.
- [7] R. B. Aronson, "Operation plug-and-play is on the way," *Manuf. Eng.*, vol. 118, no. 3, pp. 108–, 1997.
- [8] M. L. Nicora, R. Ambrosetti, G. J. Wiens, and I. Fassi, "Human–Robot Collaboration in Smart Manufacturing: Robot Reactive Behavior Intelligence," *J. Manuf. Sci. Eng.*, vol. 143, no. 3, Dec. 2020, doi: 10.1115/1.4048950.
- [9] T. R. Kurfess, C. Saldana, K. Saleeby, and M. P. Dezfouli, "A Review of Modern Communication Technologies for Digital Manufacturing Processes in Industry 4.0," *J. Manuf. Sci. Eng.*, vol. 142, no. 11, Sep. 2020, doi: 10.1115/1.4048206.
- [10] L. Ren, L. Zhang, F. Tao, C. X. Zhao Ch, and X. Zhao, "Cloud manufacturing: from concept to practice. *Enterp Inf Syst* 9 (2): 186–209," 2015.
- [11] J. Siderska and K. S. Jadaan, "Cloud manufacturing: a service-oriented manufacturing paradigm. A review paper," *Eng. Manag. Prod. Serv.*, vol. 10, no. 1, 2018.
- [12] J. J. Beaman, D. L. Bourell, C. C. Seepersad, and D. Kovar, "Additive Manufacturing Review: Early Past to Current Practice," *J. Manuf. Sci. Eng.*, vol. 142, no. 11, Sep. 2020, doi: 10.1115/1.4048193.
- [13] S. Fathizadan, F. Ju, K. Rowe, A. Fiechter, and N. Hofmann, "A Novel Real-Time Thermal Analysis and Layer Time Control Framework for Large-Scale Additive Manufacturing," *J. Manuf. Sci. Eng.*, vol. 143, no. 1, Oct. 2020, doi: 10.1115/1.4048045.
- [14] B. Kapusuzoglu, M. Sato, S. Mahadevan, and P. Witherell, "Process Optimization Under Uncertainty for Improving the Bond Quality of Polymer Filaments in Fused Filament Fabrication," *J. Manuf. Sci. Eng.*, vol. 143, no. 2, Oct. 2020, doi: 10.1115/1.4048073.
- [15] J. M. Pappas, A. R. Thakur, M. C. Leu, and X. Dong, "A Comparative Study of Pellet-Based Extrusion Deposition of Short, Long, and Continuous Carbon Fiber-Reinforced Polymer Composites for Large-Scale Additive Manufacturing," *J. Manuf. Sci. Eng.*, vol. 143, no. 7, Mar. 2021, doi: 10.1115/1.4049646.
- [16] K. L. Snapp, A. E. Gongora, and K. A. Brown, "Increasing Throughput in Fused Deposition Modeling by Modulating Bed Temperature," *J. Manuf. Sci. Eng.*, vol. 143, no. 9, Apr. 2021, doi: 10.1115/1.4050177.
- [17] L. J. Love, "Utility of Big Area Additive Manufacturing (BAAM) For The Rapid Manufacture of Customized Electric Vehicles." United States. Dept. of Energy. Office of Energy Efficiency and Renewable Energy ;, Washington, D.C. ;, 2015.
- [18] Y. Jin, H. A. Pierson, and H. Liao, "Toolpath allocation and scheduling for concurrent fused filament fabrication with multiple extruders," *IISE Trans.*, vol. 51, no. 2, pp. 192–208, 2019, doi: 10.1080/24725854.2017.1374582.
- [19] H. Shen, L. Pan, and J. Qian, "Research on large-scale additive manufacturing based on multi-robot collaboration technology," *Addit. Manuf.*, vol. 30, no. October, 2019, doi: 10.1016/j.addma.2019.100906.

- [20] N. Oxman, J. Duro-Royo, S. Keating, B. Peters, and E. Tsai, "Towards Robotic Swarm Printing," *Archit. Des.*, vol. 84, no. 3, pp. 108–115, May 2014, doi: 10.1002/ad.1764.
- [21] X. Zhang *et al.*, "Large-scale 3D printing by a team of mobile robots," *Autom. Constr.*, vol. 95, no. August, pp. 98–106, 2018, doi: 10.1016/j.autcon.2018.08.004.
- [22] D. Zhao and W. Guo, "Shape and Performance Controlled Advanced Design for Additive Manufacturing: A Review of Slicing and Path Planning," *J. Manuf. Sci. Eng.*, vol. 142, no. 1, Nov. 2019, doi: 10.1115/1.4045055.
- [23] N. Hack and H. Kloft, "Shotcrete 3D Printing Technology for the Fabrication of Slender Fully Reinforced Freeform Concrete Elements with High Surface Quality: A Real-Scale Demonstrator," in *Second RILEM International Conference on Concrete and Digital Fabrication*, Cham, 2020, pp. 1128–1137.
- [24] New York University, "Teams of mobile 3D printing robots could fix bridges on Earth and build them to Mars," pp. 1–4, 2019.
- [25] L. Poudel, Z. Sha, and W. Zhou, "Mechanical Strength of Chunk-Based Printed Parts For Cooperative 3D Printing," in *46th SME North American Manufacturing Research Conference, NAMRC 46*, 2018, vol. Accepted.
- [26] J. Mcpherson, A. Bliss, F. Smith, E. Harriss, and W. Zhou, "A Slicer and Simulator for Cooperative 3D Printing," *Solid Free. Fabr. 2017 Proc. 28th Annu. Int. Solid Free. Fabr. Symp. – Addit. Manuf. Conf.*, pp. 870–884, 2017.
- [27] Marques, Lucas Galvan, Robert Austin Williams, and Wenchao Zhou. "A mobile 3D printer for cooperative 3D printing." In *Proceeding of the 28th international solid freeform fabrication symposium*, pp. 1645-1660. 2017.
- [28] L. Poudel, C. Blair, J. McPherson, Z. Sha, and W. Zhou, "A Heuristic Scaling Strategy for Multi-Robot Cooperative Three-Dimensional Printing," *J. Comput. Inf. Sci. Eng.*, 2020, doi: 10.1115/1.4045143.