

Toolpath Planning for Multi-Gantry Additive Manufacturing

Hieu Bui, Harry A. Pierson, Sarah Nurre Pinkley & Kelly M. Sullivan

To cite this article: Hieu Bui, Harry A. Pierson, Sarah Nurre Pinkley & Kelly M. Sullivan (2020): Toolpath Planning for Multi-Gantry Additive Manufacturing, IISE Transactions, DOI: 10.1080/24725854.2020.1775915

To link to this article: <https://doi.org/10.1080/24725854.2020.1775915>



Accepted author version posted online: 29 May 2020.



Submit your article to this journal



Article views: 16

[View related articles](#) View Crossmark data 

Toolpath Planning for Multi-Gantry Additive Manufacturing

Hieu Bui, Harry A. Pierson*, Sarah Nurre Pinkley, and Kelly M. Sullivan

Department of Industrial Engineering, University of Arkansas, Fayetteville, AR 72701, USA

*Corresponding author: E-mail: hapierso@uark.edu

Abstract

Additive manufacturing (AM), specifically fused filament fabrication (FFF) is revolutionizing the production of many products. FFF is one of the most popular AM processes because it is inexpensive, requires little maintenance, and has high material utilization. Unfortunately, long cycle times are a significant drawback that prevents FFF from being more widely implemented, especially for large-scale components. In response to this, printers that employ multiple independent FFF printheads simultaneously working on the same part have been developed, and multi-gantry configurations are now commercially available; however, there is a dearth of formal research on multi-gantry path planning, and current practices do not maximize printhead utilization or as-built mechanical properties. This paper proposes a novel methodology for generating collision-free toolpaths for multi-gantry printers that yields shorter print times and superior mechanical properties compared to the state of the art. In this, a metaheuristic approach is used to seek near-optimal segmentation and scheduling of each layer while a collision checking and resolution algorithm enforces kinematic constraints to ensure collision-free solutions. Simulation is used to show the resulting makespan reduction for various layers, and the proposed methodology is physically implemented and verified. Tensile testing on samples printed via the current and proposed methods confirm that the proposed methodology results in superior mechanical properties.

Keywords: Fused filament fabrication; toolpath planning; parallel processing; multi-gantry 3D printer

1 Introduction

Additive manufacturing (AM), also known as 3D printing, is a process of creating three-dimensional objects one layer at a time based on a computer-aided design (CAD) model. AM can produce complex geometric objects that are unachievable or require combinations of traditional manufacturing processes. Fused filament fabrication (FFF) is a specific AM process that creates a workpiece by extruding a thermoplastic polymer. FFF is widely available and used in many applications because of its affordable cost and ease of use. The applications include prototypes, customized functional models, tooling and moldings, and end-use parts (Cozier et al., 2015). Since it was first introduced in the 1980s, there have been many significant improvements on FFF technology. These improvements have many aspects from faster printing speed to the use of a wide array of printable materials (Compton and Lewis, 2014; Li et al., 2016; Lee et al., 2017).

Despite these improvements, real-world application of FFF remains constrained by the fundamental tradeoff between material deposition rate and geometric resolution. This makes FFF slow, and severely limits the potential to scale the process up to large workpieces. A handful of multi-agent FFF solutions have been proposed in the literature (Zhang et al., 2018; Greene, 2014; Marques et al. 2017), where multiple FFF extruders work simultaneously on a single workpiece. In particular, the concept of a multi-gantry printer was introduced by Autodesk, with their prototype developed under the name Project Escher (Atwell, 2016). This design was subsequently commercialized by Titan Robotics (Titan Robotics, 2017) as the Cronus 3D printer.

The advent of multi-agent printers introduces a complex planning problem. The toolpath segments required to create a workpiece must be segregated among the various printing tools, and the execution of these segments must be ordered and scheduled to both satisfy process constraints and avoid collision between tools. In general, these tasks are interdependent, giving rise to a difficult

simultaneous segregation and scheduling problem. The current state of the art divides the work between tools in a manner that makes scheduling trivial. This separates and simplifies the segregation and scheduling problems, but at the expense of efficiency and mechanical properties. The predominant method, referred to as the orthogonal segmentation approach (OSA) in this paper, is mostly closely associated with multi-gantry printers, but it is representative of the “chunking” approach taken by other systems. OSA divides each layer into sub-regions along lines orthogonal to the axis of gantry travel, as shown in Figure 1a. Each gantry prints its assigned sub-regions in a left-to-right sequence, thereby avoiding the collision. This method has two key drawbacks: 1) It results in seams between sub-regions that are generally not aligned with the infill raster direction, compromising the strength of the part due to the anisotropic mechanical properties inherent in the FFF process. This effect is especially detrimental when printing polymer matrix composite materials (Pierson et al., 2019; Hmeidat et al. 2018). 2) Simulation studies reveal that OSA results in sub-optimal printhead utilization, meaning that the benefit of the multi-gantry printer is not fully realized (Bui et al., 2019; Bui, 2019).

To mitigate these drawbacks, this paper considers the simultaneous segregation and scheduling problem for a two-gantry printer. To the best of the authors’ knowledge, this is the first published work to propose a solution that addresses the interdependence of the segregation and scheduling problems. A metaheuristic approach is proposed to automatically find feasible solutions that maintain the infill raster throughout the part, with the objective of minimizing the total time to print each layer (makespan). Within each iteration of the heuristic, a novel collision detection and resolution subroutine enforces the kinematic constraints of the printer by intelligently introducing delays in the current toolpath solution to ensure collision-free execution. The method is tested in both simulation and hardware, and the results are compared to OSA with regard to makespan and mechanical properties.

Main Contributions. The main contributions of this work are as follows. 1. It is the first to consider simultaneous segregation and scheduling for multiple gantry additive manufacturing. 2. It is the first to design a Tabu Search procedure with a novel collision checking subroutine to minimize the time needed to print each layer. 3. The developed solution method was validated on a custom 2-

gantry printer. It was found to prevent collisions, and the experimental results are similar to the computer simulation.

2 Literature review

Recent advances in the FFF printing process combined with the increased availability of consumer-grade 3D printers have motivated numerous research efforts. We proceed by outlining the relevant research in toolpath planning, robotics toolpath planning, multi-gantry printing, and collision avoidance methods. With each, we describe the current research capabilities and limitations. We proceed first, by describing the typical problems and models for toolpath planning, specifically, the traveling salesman problem and the undirected rural postman problem, and the heuristic solution methods.

Traveling salesman problem (TSP) is one of the most widely study combinatorial optimization problems. Given a set of cities and the cost of travel between each pair of cities, the TSP tries to find the cheapest route to visit all the cities exactly once and returns to the starting point. The routing of the tool can be formulated as the TSP (Castelino et al., 2003; Oysu and Bingul, 2007; Volpato et al., 2014; Lin et al., 2015; Ganganath et al., 2016).

The undirected rural postman problem (URPP) is defined on a graph $G = (V, E)$, where V is the set of vertices, and E is the set of edges. The URPP seeks a minimum cost route, where cost could represent distance, time, which visits all the required edges $E_r \subseteq E$ at least one. The prefix undirected means that each edge can be traversed in both directions. Researchers have also used the URPP for toolpath planning, where the tool must traverse all the required edges (Williams and Burdick 2006; Rodrigues et al. 2001; Fok et al., 2018).

To determine effective toolpath plans for FFF process, most existing research considers the optimization problem as a variant of the TSP or the URPP. In this context, cost could represent distance, time, or repositioning movements. As these problems are proven to be NP-hard, various heuristic and approximation algorithms have been applied to solve them. Volpato et al. (2014) solve a TSP-based toolpath planning problem using two heuristics: a nearest-neighbor procedure and a combination of the nearest-and-farthest-insertion method. In this, they found considerable benefit due to applying optimization. Fok et al. (2018) apply an extended ant colony optimization (ACO)

algorithm on a URPP-based toolpath planning problem, and Wah et al. (2002) apply both a genetic algorithm (GA) and integer programming subroutines to solve an asymmetric-TSP-based toolpath planning problem. Wojcik et al. (2015) apply a GA to solve a toolpath planning problem, where a so-called “modified zig-zag algorithm” is used to generate tool subpaths that are then synthesized within the GA. While these studies demonstrate the benefit of applying optimization to toolpath planning for FFF, they consider only machines with a single printing tool. In multi-gantry FFF, however, the possibility for collision arises due to multiple printheads working in proximity, which changes the optimization problem substantially. Thus, there is a need to extend the research base in FFF optimization to multi-gantry printing configurations.

In addition to 3D printing, toolpath planning is covered extensively in robotics literature, especially in computer numeric controlled (CNC) machining applications. The general objective of CNC machining is to achieve a balance between the quality of the machined surface and the machining time (Chen et al., 2004). To reduce machine time, the optimization techniques and methods focus on minimizing unproductive airtime during machining. Oysu and Bingul (2007) proposed a method to find a near-optimal sequence of drilling path for hole-cutting operation. They formulated the cutting toolpath as a special case of TSP and applied a GA to solve it. Koenig and Jouaneh (2005) applied optimization to generate toolpaths for cutting and welding applications, yielding a reduction of more than 12% of the total cutting and air distance. However, the existing toolpath planning optimization literature is available only for single tool CNC applications and is therefore not directly applicable to multi-gantry FFF.

Concurrent 3D printing is still a relatively new concept in AM. Jin et al. (2017; 2019) developed two heuristics to allocate toolpaths, with collision constraints, created by existing slicing software to available printheads. The simulation results for three printheads showed as much as a 60% reduction in printing time compared with single printhead printers. The method also demonstrated a technique to determine the optimal number of printheads for a specific layer. Zhang et al. (2018) demonstrated the use of multiple mobile robots that work in a team to print a large concrete structure for the building and construction industry. Experimental results showed that the system is flexible and has high scalability and efficiency. Choi et al. (2010) proposed an approach to concurrent toolpath

planning for a multi-material layered manufacturing process. The approach applied a dynamic priority scheme to adjust the motion of the tools when a potential collision is detected. While early investigation on concurrent toolpath planning for AM has been proposed, a dearth of published research on this subject represents a significant knowledge gap that is limiting AM's potential.

The problem of coordinating multiple moving objects that work in a shared workspace, without collision, is covered in various applications such as mobile robots, manipulation of robot arms, and so forth (Tsai-Yen and Latombe, 1995; LaValle and Hutchinson, 1998; Lee and Kim, 2006; Wagner et al., 2008). In each application, various methods have been introduced in which avoiding collision is a significant component. The similarity of these methods is that the potential collisions are detected at the beginning, and then the motions of corresponding objects are subsequently coordinated to avoid the collision. There are two categorizations of approaches to solve a multi-object motion planning problem: (1) coupled methods, and (2) decoupled methods. In coupled methods, the configuration spaces of all objects are combined into one unified configuration space, and the feasible path is then searched. On the other hand, in the decoupled method, the path of each object is individually created and subsequently organized to avoid collision (Peasgood et al., 2008). Several decoupled methods have been introduced, such as adjusting the trajectory, inserting time delay, and modifying the velocity of objects (Todt et al., 2000). However, in the concurrent FFF process, the printheads do not have a collision-sensing capability, self-control, or a way to communicate with each other. They are controlled by a central controller where all actions are predefined before executing. Furthermore, the printhead should not vary its speed as it could print an inconsistent road width, which allows air voids to get trapped inside compromising the structural integrity of the 3D printed part. Therefore, planning collision-free trajectories in real-time is not ideal in concurrent FFF. Adjusting the toolpath and inserting time delay when generating toolpath for each printhead provide a strong foundation for developing the subroutine to address the collisions.

In general, the current literature on path planning optimization for FFF applies various heuristic techniques to minimize the total repositioning distance. However, it focuses only on a single toolhead configuration, and hence no strategy is required for collision-avoidance. The literature on multi-object path planning is covered extensively but does not adequately provide information for

solving the concurrent FFF path planning problem. The concurrent 3D printing process is still a new concept and requires more development. This gap motivates the development of

- New toolpath planning method for multi-gantry FFF printers where the objective is to minimize makespan subject to collision constraints while achieving good mechanical properties.
- The optimization procedure that uses a combination of heuristic technique and collision avoidance subroutines to obtain near-optimal toolpath solutions for a multi-gantry printer.

3 Method

3.1 Overview

There are traditionally two factors that contribute to the printing time, also referred to as makespan: (1) the time to complete printing movements, and (2) the time to complete rapid movements between printing. In multi-gantry FFF printing, gantries may have to wait or make additional non-value-added movements in order to avoid a collision. In order to minimize the makespan of a layer with respect to the aforementioned factors, a tabu search (TS) heuristic was selected to optimize each printhead's toolpath. To evaluate solutions in TS and ensure the toolpaths created from these solutions are collision-free, two subroutines were developed: (1) collision checking and resolution - thus the name TS-CCR. The proposed method was developed under several assumptions.

- (1) A two-gantry configuration was selected. In this paper, the left and right gantries refer to the position of them on the machine.
- (2) The perimeters for the entire model were assumed to be printed by one printhead. The rationale of this assumption is to eliminate the discontinuities along the outer perimeter (i.e., the location where outer perimeters of two adjacent subregions touch in the OSA approach), allowing the printed model to have a better surface finish (Netfabb, 2018).
- (3) Under the multi-gantry configuration, the gantries travel and collide only in the x-direction. This assumption is especially important in the development of the two subroutines where the collisions of gantries are detected and resolved.

3.2 *Tabu search heuristic*

The proposed method can be broken down into two problems. Given a number of infill raster segments (Figure 2), the first problem is to assign these segments to each printhead, and the second problem is to sequence the segments on the assigned printhead to avoid mutual collision (sequencing problem). TS is selected because it has been proven to be effective on a wide variety of optimization problems, including TSP (Glover, 2001). TS stores information in memory, then uses it to guide and restrict the future search in order to obtain a better solution and to overcome local optimality.

The representation of a solution in TS is comprised of a raster segment list corresponding to each printhead. The list associated with a given printhead specifies the order in which assigned raster segments of that printhead are printed. To evaluate the quality of the solution, the CCR subroutines, which will be discussed in the next section, are used. Specifically, the result from the CCR provides the infill makespan (whereby a smaller makespan means a better solution). Three operators are designed for the TS to generate new solutions. Specifically, the first two operators are designed to allow TS to improve the current solution by modifying the (1) assignment and (2) sequencing decisions. An illustration of these operators is shown in Figure 3. First, the global swap (GS) operator exchanges two raster segments from the sequences of different printheads. This operator is crucial in deciding the best raster segment assignment for each printhead. Second, the local swap (LS) operator is used to swap two raster segment orders for the same printhead. By changing the order in which the raster segments are printed, the LS aims to reduce the number of rapid movements required, also reducing the makespan. Because of the changes caused by the first two operators, the difference of the makespan of the two printheads can be significant; thus, the last operator is designed to rebalance the makespans. The rebalancing operator is used to distribute one raster segment from the sequence that results in larger makespan to the one with smaller makespan. This raster segment is added to the end of the new sequence (i.e., last to print).

At the beginning of the algorithm, raster segments are randomly assigned to each printhead. The tabu search heuristic does not explicitly prohibit solutions with collisions but instead allows the CCR to resolve them by imposing delays within each printhead's sequence of raster segments. The

infill makespan of the random initial solution is expected to be large because of the chance for significant delays (introduced by CCR to avoid collisions) and/or lengthy rapid movements.

One operator is selected in each iteration based on a user-specified probability. At the early stage of the TS, the probability that the GS is selected is greater than the LS. After each iteration in which the GS cannot find an improved solution, its probability is slightly reduced. As the search continues, the LS operator slowly becomes more attractive. In addition, the probability of the TS choosing the rebalancing operator is set relatively small. This means that the search occasionally checks for the opportunity to balance the infill makespan between the two toolpaths instead of doing so in every iteration.

A random subset of moves within the selected operator is sampled. The new solutions resulting from the sampled moves are stored in a candidate list. The CCR is then applied to evaluate the makespan for each solution in the candidate list, and the TS selects one that improves upon the current solution's makespan. If no move reduces the makespan, the TS selects the solution with the least increase in makespan. This helps the search process reach different regions of the search space. Every time a solution improves upon the best-known solution, it will be updated as the best-known solution. The move that created the selected solution is then labeled as a tabu in the short-term memory list, called tabu list. Tabu lists are used to prevent cycling by preventing solutions that contain tabu moves. The moves in the tabu list can be removed using the first-in-first-out method. One tabu list is designed for each of the three operators. A simple aspiration criterion is used, specifying that a tabu move may be selected if it results in a solution with a smaller makespan than the current best-known solution.

Two stopping criteria are used to terminate the TS. First, the TS terminates if the elapsed time at the end of an iteration exceeds a user-defined time limit. Second, the TS also terminates if the best-known solution is not improved above a certain threshold within another user-defined time limit. For example, if after a new best-known solution is identified, the makespan improves by less than 1% in the subsequent 5 minutes, the TS terminates. Figure 4 represents a flowchart of the proposed TS procedure.

3.3 Collision checking and resolution subroutines

In the proposed method, the CCR is used to evaluate the makespan of the candidate solution. Figure 5 illustrates the flowchart of the solution evaluation process.

The toolpath for each printhead is first constructed from the solution. The toolpath conversion procedure starts from the beginning of the sequence, then depending on the location of the next raster segment, different types of connectors are chosen. After the printhead finishes printing a segment, the current segment's endpoint will be connected to the closest point of the next sequenced raster segment. If the next segment is adjacent to the current segment, the endpoint of the current segment will be connected to the closest point of the adjacent segment with a solid connector. This means the printhead will continue to dispense the filament when moving to the next segment. Else, if not adjacent, the two endpoints will be connected with a dashed connector, meaning the printhead will stop extruding the filament and perform a fast maneuver to the next segment. This assumption influences the printing direction of each print segment. The procedure continues until all the segments are visited. Figure 6 illustrates the toolpath of a potential solution for a two-gantry configuration on a notional example and TS solution. The representation of the TS solution in this example is

$$s = \begin{bmatrix} 33, 36, 39, 43, 47, 50, 53, 57, 62, 46, 40, 37, 34, \\ 31, 27, 25, 22, 28, 21, 18, 15, 12, 9, 11, 5, 6, \\ 2, 0, 7, 3, 4, 8, 10, 13, 16, 19, 24, 30, 1 \\ 65, 69, 71, 73, 76, 79, 82, 86, 87, 89, 94, 97, 91, \\ 93, 90, 88, 85, 84, 81, 70, 66, 61, 60, 56, 55, 52, \\ 49, 44, 45, 42, 41, 38, 35, 32, 29, 26, 23, 20, 17, \\ 14, 48, 51, 58, 64, 54, 59, 63, 67, 68, 72, 74, 75, \\ 78, 77, 80, 83, 95, 92, 96, 98 \end{bmatrix}$$

In this example, the toolpath of the left printhead started moving from the park position (0, 0) to the upper-left corner of raster segment #33, iteratively moved through the raster sequence until it reached segment #1, and finally moved back into the park position. Similarly, the right printhead started from the park position (1800, 0) to the upper-left corner of segment #65, and iteratively moved through the sequence ending at segment #98 and back into its park position.

It is important to separate the two different types of connectors because they have different operating speeds (printing speed vs. rapid movement speed). The time associated with completing each linear movement in the toolpath is then calculated using a trapezoidal velocity profile. In

particular, each movement has an entry speed, acceleration, operating speed, and deceleration values. At each movement, the printhead begins to accelerate from the entry speed to the operating speed at a constant rate of acceleration. It then moves at the target speed for a certain distance and decelerates to the exit speed (i.e., entry speed of the next segment) at a constant rate of deceleration. The acceleration, deceleration, and jerk parameters are machine-specific and need to be specified accordingly in order to ensure the estimated makespan is accurate. Note that the “jerk” used in most 3D printer’s firmware, Marlin, is not the same as the “jerk” term in physics. The jerk, in this case, denotes the maximum instantaneous velocity change (i.e., measured in mm/s instead of mm/s³). This value is used to determine the speed at the junction of two movements (also described as the exit speed of the current movement and the entry speed of the next movement). Without jerk, the printhead needs to perform a complete stop in every corner, because it cannot begin accelerating in the new direction before arriving at the corner. Because the printhead cannot make instantaneous velocity change, the jerk will cause some vibrations to the machine. However, these vibrations are negligible compared to the reduction in the printing time.

Because the gantries collide only in the x-direction, the collision-checking subroutine is straightforward: a collision happens when two gantries share the same workspace at any moment in time. A trajectory plot of the adjusted x-coordinates of the gantries versus time can be constructed to visualize collisions. A user-defined safety distance between gantries is added to provide a buffer zone. The x-coordinate of each gantry is calculated by offsetting the x-coordinate of the printhead by θ . The equations with illustrations (Figure 7) of the x-coordinates of the left and right gantries are:

$$x_{LG} = x_{LP} + \theta, \quad (1)$$

$$x_{RG} = x_{RP} - \theta, \quad (2)$$

$$\theta = \frac{(\text{gantry width} + \text{safety distance})}{2}, \quad (3)$$

where x_{LG} and x_{RG} are the x-coordinates of the left and right gantries, respectively, and x_{LP} and x_{RP} are the x-coordinates of the left and right printheads, respectively.

The collision checking subroutine either (1) concludes that the two gantries’ trajectories are collision-free, or (2) identifies the first collision in time. The collision resolution subroutine then

resolves the first collision by introducing delay, but by doing so, it might create more collisions at later times. Thus, by alternatively using both subroutines, they can remove all collisions.

The CCR is designed to be computationally inexpensive as it is frequently used throughout the TS. Figures 8 and 9 illustrate the trajectory plot of the toolpaths after performing the collision checking subroutine and the final trajectory plot after performing the CCR of the toolpaths in the above example.

3.3.1 Collision checking subroutine

The trajectory is comprised of paths that gantries follow in the x-direction as a function of time. Each gantry's trajectory, which is used in the CCR, can be expressed as a piecewise linear function whose line segments (in x-time space) correspond to the raster segments and rapid movements of the printhead. Feasible configurations must prohibit the gantries from occupying the same x-coordinate or passing through each other. With respect to the trajectory plot, this means that the right gantry trajectory must not fall below (i.e., its x-coordinate become smaller than) the left gantry trajectory. The collision-checking subroutine starts from selecting the earliest segments at time $t = 0$, one from each gantry. Antonio's algorithm (Antonio, 1992) is applied to check whether the selected pair of segments intersect. If there is an intersection, the associated segments are recorded as well as the intersection point. Otherwise, the trajectory segment with the earlier finishing time will be replaced by its next segment. The checking process continues until all pairs of segments are checked, or the first collision is detected.

3.3.2 Collision resolution subroutine

The collision resolution subroutine introduces delays into the trajectories of the gantries to avoid potential collisions. When the first collision is identified, the collision checking subroutine provides two outputs: (1) the set of intersection points $(t_{\text{start}}, x_{\text{start}})$ and $(t_{\text{end}}, x_{\text{end}})$ of the first collision, and (2) the pair of segments that cause each intersection.

Let I^l and I^r respectively represent the set of vertex indices of the left and right trajectories, i.e., such that the vertex sets $\{(t_i^l, x_i^l) | i \in I^l\}$ and $\{(t_i^r, x_i^r) | i \in I^r\}$ represent the breakpoints in the

piecewise-linear trajectory of the left and right gantries. Let P^l and P^r represent the vertices of the left and right trajectories whose t-value is in the interval $[t_{\text{start}}, t_{\text{end}}]$. Let CI^l and CI^r denote the set of vertex indices of P^l and P^r such that the following equations are satisfied:

$$P^l = \{p_i^l = (t_i^l, x_i^l) \mid i \in CI^l\} \quad (4)$$

$$P^r = \{p_j^r = (t_j^r, x_j^r) \mid j \in CI^r\} \quad (5)$$

In order to determine where the delay can be inserted, the subroutine starts by looking at the vertices before the identified collision and selecting one at which imposing a delay would resolve the collision. To eliminate the collision, a delay may be imposed on the left gantry at any vertex $p_L^l = (t_L^l, x_L^l)$, $L \in I^l$ for which $x_L^l < x_j^r, \forall j \in CI^r$, and $t_L^l < t_{\text{start}}$. Thus, by allowing the left gantry to wait long enough at the vertex p_L^l , the right gantry can complete its trajectory up to t_{end} without conflict. Among all values of L satisfying this characterization, let L denote the one with the largest value of t_L^l . Similarly, let R denote the one with the largest value of t_R^r such that $x_i^l < x_R^r, \forall i \in CI^l$, and $t_R^r < t_{\text{start}}$. By delaying the right gantry at the vertex p_R^r until the left gantry has completed its trajectory up to the time t_{end} , the conflict can be resolved. To ensure such vertices p_L^l and p_R^r always exist; the left gantry must be positioned at the machine's minimum X location before executing the print job, while the right gantry at the maximum. This is illustrated in figures 6, 8, and 9 where the left printhead is parked at position (0, 0) while the right printhead is parked at position (1800,0).

Let d^l and d^r respectively represent the amount of delay needed to add to p_L^l and p_R^r to resolve the collision. These distances can be determined as

$$d^l = \max \{t_{\text{dist}}(p_i^l, \overline{p_j^r p_{j+1}^r}) \mid \forall i \in CI^l, \forall j \in CI^r \setminus \max(CI^r)\} \quad (6)$$

$$d^r = \max \{t_{\text{dist}}(p_j^r, \overline{p_i^l p_{i+1}^l}) \mid \forall i \in CI^l \setminus \max(CI^l), \forall j \in CI^r\} \quad (7)$$

where $t_{\text{dist}}(p, \overline{uv})$ represents the horizontal distance in the time axis (i.e., amount of delay) from vertex p to the segment \overline{uv} . An illustration of $t_{\text{dist}}(p, \overline{uv})$ is shown in Figure 10.

The subroutine then picks the vertex that requires the least amount of delay (i.e., either a delay of d^l imposed at vertex p_L^l or a delay of d^r imposed at vertex p_R^r) to resolve the collision. A small user-defined epsilon (default was set to 0.2s) is also added to the delay to create a separation in

the time axis as shown in the zoom section of Figure 9. The trajectory segments following the selected vertex are adjusted by shifting them to the right by d seconds. Figure 10 illustrates the procedure of the collision subroutine to solve a simple collision.

$$d = \min(d^l, d^r) + \epsilon \quad (8)$$

3.4 Closed-loop control and resynchronization process

For two-gantry 3D printers in this research, the closed-loop control system is developed to keep track of the x-position of the gantries. It ensures the gantries operate at their intended trajectories and avoid collisions. Rotary encoders are mounted to the stepper motors that move the gantries.

The proposed system requires each gantry to have a dedicated controller, meaning that each gantry needs a separate G-code file. The computer sends the G-code commands line-by-line to the controller until its queue is full. To generate the G-code file for each gantry, the associated toolpath is extracted and converted to a series of G-code commands. Even though the proposed method utilized the trapezoidal velocity profile to model the trajectory of the gantries, the actual trajectory might be slightly different. To resolve this issue, a resynchronization process was designed. The process analyzes the data received from the encoders and determines when the two gantries need to resync to ensure these gantries follow their intended trajectories. By manipulating the G-code sending process, the resync action can be added during the print. Each resync action is comprised of two G-code commands, M400 and G4. In Marlin firmware, when the machine's controller receives the M400 command, it will stop accepting new commands and wait until all the moves in the queue are finished. The G4 command is used to pause the machine, the gantry in this case, for a specific amount of time.

The CCR ensures that the trajectories of the gantries do not overlap; this means that there is always a safe margin between them. When the observed separation between two gantries is smaller than the predefined safety margin, the process lets the gantries resync before continuing. Several Python scripts are created: (1) sending G-code commands to each gantry, (2) reading the data from encoders and identifying opportunity for adding resync action, and (3) keeping track of the state of completion of the resync action for the gantries. In the developed process, the state in the last script prevents all gantries from accepting new commands until their queues are cleared. Upon the

completion of each resync action, a small delay, represented in the G4 command, is added to account for the difference between the starting time of the next trajectories. Figure 11 illustrates the flowchart of the closed-loop control with the resync process for the two-gantry configuration. In order for the resync process to be successfully implemented, the aforementioned Python scripts need to run concurrently. The multiprocessing package in Python allows modern computers to accomplish such a requirement.

3.5 Experimentation

The purpose of the simulation experimentation is to compare the performance of the proposed TS-CCR method with the OSA. This is done on a selected layer of four different 3D CAD objects, each with different layer complexity. These layers' geometries represent some of the practical uses that would benefit from multi-gantry 3D printers as the size of these layers is relatively large, thus requiring significant printing time to complete. The physical experimentation includes two main objectives: The first is to verify that the optimized toolpath is collision-free when implementing it on a custom two-gantry printer, the second objective is to show the mechanical properties of the finished objects using the TS-CCR are at least on par when compared to the OSA.

3.5.1 Simulation setup

For each object, a layer of 0.3 mm is sliced with three different approaches and their makespans are compared. The makespans from these approaches are as follows:

- (1) The theoretical minimum makespan is calculated by dividing the makespan of a single printhead printing by two.
- (2) The OSA makespan is calculated by slicing the layer using the Netfabb Multi-gantry FFF Engine plugin.
- (3) The TS-CCR makespan is calculated by adding infill makespan from the TS-CCR to the makespan of printing the perimeters using one printhead.

The selected layers from 3D CAD objects are shown below:

The Titan Cronus multi-gantry printer profile in the Netfabb software was selected as the default value for the printed build volume, at 1900 mm x 750 mm x 450 mm. The gantry width of the

printer was set to 126 mm with 150 mm of the safety distance. The printing speed was set to 50 mm/s, the rapid travel speed was set to 80 mm/s. The acceleration, deceleration, and jerk were set to 2000 mm/s², 2000 mm/s², and 8 mm/s, respectively. The infill percentage was set to 30%.

For the TS heuristic, the size of the candidate list was set to 10. The tabu list size for each operator was chosen as 5. The probabilities for global swap, local swap, and rebalancing operators were initialized to 0.7, 0.2, 0.1, respectively. The elapsed time was limited to 7 minutes. If the TS cannot find a solution with over 2% improvement in 3 minutes, the algorithm terminates and returns the best-known solution.

3.5.2 Custom multi-gantry printer

Because multi-gantry FFF technology is relatively new, the availability of the printers that are designed to run this technology is still limited. To physically verify that the toolpath from the proposed method is collision-free, a custom two-gantry printer was designed, as shown in Figure 13a. The y-axis and z-axis structures were adopted from two Creality Ender 3 3D printers. The x-axis was reworked so that each gantry can move independently from each other, as shown in Figure 13b. The rotary encoder is mounted next to the x-axis stepper motor to measure the position of each gantry. The build surface of the custom machine measures 470 x 235 mm.

3.5.3 Tensile specimen design and testing

In order to fit into an MTS machine, small specimens are designed and printed with a custom profile for each printing approach. Instead of using two printheads to concurrently print different portions of the specimen, one printhead was utilized to print these portions in sequential order. Each specimen is comprised of two parts: a 3D printed part and four steel tabs. Tabs are used to prevent gripping damage to the specimen. The 3D printed parts are printed with Polylactic Acid (PLA), and the steel tabs were glued to the printed parts using J-B Weld plastic bonder. The printing parameters were kept the same for both printing approaches; they are listed in Table 1.

The OSA divided each layer into two sub-regions with the seam parallel to the y-axis (Figure 14). This created a notable seam in the printed parts where each sub-region required its own perimeter shells. To hide and strengthen the seam, the OSA approach randomly offsets the seam in each layer.

The ASTM D3039 standard was used to measure the tensile properties of polymer matrix composite materials. Tensile testing measurement procedures were conducted on the MTS testing machine with mechanical wedge grips attached. Ultimate tensile strength was chosen to compare the mechanical properties of the two toolpath planning approaches.

4 Results

4.1 Simulation results

Three types of makespans were calculated to compare the performance of the proposed TS-CCR method and OSA. They are (1) theoretical minimum makespan, (2) the OSA makespan, and (3) the proposed method makespan. Figure 15 illustrates the comparison chart between the three makespans. In all four selected layers, the proposed methodology produces solutions with smaller makespans than the OSA and helps bring the overall makespan closer to the theoretical minimum. The percentage improvement varies depending on the complexity of the layer. The proposed method reduces the makespan by 15.14% on the simple Airfoil frame layer, while the improvement reduces to 7.72% for the “IE Hog.”

4.2 Physical experiment results

4.2.1 Results from tensile testing

On average, the UTS of the TS-CCR specimen (41.31 MPa) was found to be stronger in comparison to the UTS of the OSA specimen (30.64 MPa). Table 2 shows that the UTS standard deviation of the TS-CCR specimens is smaller than the other. A two-sample t-test was conducted to confirm that the two UTS means are different. The P-value of 0.013 indicates that the two UTS results are statistically significant at the 5% significance level.

Figure 16 illustrates the break behavior of the specimen printed with the TS-CCR and the OSA. OSA's specimens show a brittle fracture. All OSA samples exhibited brittle fracture and broke at and along the seam. The TS-CCR samples broke at different points along the gauge length.

4.2.2 TS-CCR and closed-loop control validation

The “IE Hog” with 100% infill density was chosen to validate the collision-free toolpaths and the performance of the closed-loop control system. Figure 17 shows the complete layer that was printed on the custom printer. Specifically, the red portion was printed by the left gantry, while the blue portion was printed by the right gantry. Note that only the infill toolpaths were printed.

One resync action was added to the G-Code files at the beginning of the actual printing that allows both gantries to go to their starting positions and begin their jobs at the same time. However, this action was not counted in the comparison. The encoders began to read the position when the gantries started the homing process. In total, two resync actions were needed during the printing process to allow each gantry to follow its intended trajectory. This means that the closed-loop control and resynchronization process were able to identify and resolve the minor differences in the calculated and actual trajectories. The actual printing with two resync actions took 1,346 seconds to complete, 3 seconds more than the calculated makespan, as shown in Figure 18.

5 Discussion

5.1 Performance of TS-CCR

As shown in section 4, the TS-CCR yields better solutions in terms of makespan when compared with those obtained from OSA. The proposed method can intelligently, without the help of the user, assign the raster segments to each gantry in a way that does not create a collision. There are primarily two factors that can affect the performance of the proposed method, namely (1) layer features, and (2) process parameters.

Layer complexity plays an important role in determining the performance of the TS-CCR. Because the proposed method was designed only to optimize the infill aspect of the layer, the perimeters are printed using only one gantry. This means for any complex layer that requires significant time to print the perimeters; it would offset the improvement that the proposed method could achieve. For example, the “IE Hog” layer contains several irregular shapes that require the gantry to spend a long time to print these perimeters, thus increasing the overall makespan. Future

research on a strategy that allows multiple gantries to print the perimeters simultaneously could be implemented to reduce the overall makespan even further.

As mentioned in the method section, the gantry movements in the x-direction must be carefully planned to avoid collisions. Thus, the x-dimension of the 3D model is the most critical factor that determines whether that model can be printed on the multi-gantry 3D printer using the proposed method. Two features of the layer that affect the performance of the proposed method are (1) the aspect ratio between the x- and y- dimension of the layer, and (2) the ratio between the x-dimension of the layer and the width of the gantry. The proposed method performs well when these two ratios are large, as shown in the “airfoil frame” layer. As these ratios decrease, the chance for the gantries to collide is increased. This means the CCR is expected to introduce more delays to resolve all the collisions, thus reducing the effectiveness of the method. When the value of the ratio between the x-dimension and the gantry’s width is too low, the gantries have little room to operate. The OSA might fail to produce any result while the TS-CCR might produce the result with a larger makespan than the single printhead printing approach. In this case, the whole layer will be printed by one gantry. These two ratios can be adjusted by rotating the 3D model on the z-axis. Thus, the orientation and the size of the 3D model must be analyzed before applying the proposed method.

Safety distance was determined by the user. There is currently no method to define the optimal safety distance value, but it can be done via user expertise or by a trial-and-error approach. Like the gantry width, the safety distance limits how much each gantry can move without registering a collision. A good starting value can be set to approximate the gantry width (i.e., 150 mm for the safety distance vs. 126 mm for the gantry width). Note that the gantry width value is fixed for a machine, while the safety distance can be adjusted. Increasing this value requires the TS-CCR more time to run to find good solutions.

In TS, each solution is evaluated by the CCR. Although the CCR can evaluate the solution relatively quickly (~300 ms for a given solution), the large number of solutions results in significant computation time. Thus, the size of the candidate list was found to be the most important TS parameter that affects the computational time of the proposed method. Depending on the different layers’ geometries, the size of the candidate list is required to be adjusted. The initial probability of

each operator was also found to be an important parameter in TS. As the TS is executed, the need for the global swap is diminished in favor of local swap because the number of raster segments begins to allocate well to each gantry. These values determine how fast the TS is switching from choosing the global swap to the local swap in each iteration. Another aspect of the process parameters can be expressed as the infill percentage. As the infill percentage increases, the number of raster segments increases. The TS-CCR is expected to require additional time to find a good solution. However, the computing time to optimize each layer is significantly smaller than its makespan. For example, the TS-CCR on each layer was limited to 7 minutes while the makespan is 26 minutes for the “airfoil frame” layer. One can argue that it is computationally expensive to optimize every layer before sending them to the machine. One possible way to mitigate the speed limitation is allowing the machine to execute the current layer n while spending time performing TS-CCR on the next layer $n + 1$. The advantage of this is that allowing the TS-CCR to run for a longer duration on each layer can potentially generate better solutions. Another improvement to the TS-CCR can be made by improving the efficiency of the collision resolution subroutine. The current subroutine resolves the collision by adding a simple delay to one of the gantries. This means the gantry stops and waits until the toolpath is clear to execute. However, instead of staying in one place while waiting (i.e., no change in the gantry’s x-coordinate in the trajectory plot), the gantry could potentially move itself out of the way and let another gantry continue to print. This provides extra flexibility when dealing with collisions, thus reducing the number and duration of the delays needed. A new collision resolution subroutine will need to be developed to account for the new change.

The methodology proposed in this paper is flexible and can be useful in other configurations beyond multi-gantry. One of the AM technologies that currently gains a lot of attention is multi-laser selective laser sintering (SLS). There are three potential issues when using multiple lasers in proximity which lead to degradation of mechanical properties: (1) the airborne condensate from one laser could reduce downwind laser spot intensity, (2) the airborne spatter and condensate could obscure the laser from reaching the powder bed, and (3) large spatter particles could be incorporated in the component (Saunders, 2018). In place of the collision constraint in multi-gantry FFF, a constraint on laser assignment can be developed. One constraint could be that the distance between

the concurrent scan segments must satisfy some predefined distance rules to mitigate the aforementioned issues and avoid the interaction between lasers. The TS algorithm can then be used to optimize the sequence in which all scan segments are printed. This can help the process achieve a balanced workload for all lasers while attaining good mechanical properties.

5.2 Physical test

The closed-loop control system and resynchronization process were developed to adjust the input of G-code commands to reduce the error between the actual trajectories and the desired ones. Also, the closed-loop control system in this research was desired to only keep track of the movements in the x-axis as a function of time. This means the system did not consider the missed step issue. For example, if the one stepper motor loses steps during a print, then all the layers following it get misaligned, and this results in a failed print. A system that allows the firmware of each gantry to adjust the printhead target position when the motor loses steps has not been investigated. Even though this issue is uncommon, it is unrecoverable under the developed system. To ensure that the gantries do not collide when this problem arises, the M112 command can be utilized to immediately shut down and prevent damaging the machine.

As shown in the Results section, all the OSA specimens broke at the seam due to the initial separation between the different sub-regions' perimeters at one layer. Because the layer sliced by OSA generally has various seams, there is an increased chance for the layer to fail under stress. Because the proposed method utilizes one gantry to print the perimeters resulting in no seam, the printed part achieves better mechanical properties.

6 Conclusion

A new toolpath planning methodology, TS-CCR, has been developed to generate the collision-toolpath for the two-gantry FFF printer. The TS-CCR has been shown to provide solutions with shorter makespan than the available approach, OSA, while achieving good mechanical properties. The toolpaths of one selected layer were printed on the custom two-gantry printer. It helped validate that the TS-CCR is capable of producing collision-free toolpaths. A closed-loop control system and resynchronization process were developed to monitor the execution of the TS-CCR toolpaths. The developed control system demonstrated the ability to detect the differences between the calculated

and observed trajectory plots and use the resync action to correct them.

Notes on Contributors

Hieu Bui earned his B.S. and M.S. degrees in industrial engineering from the University of Arkansas – Fayetteville in 2015 and 2019, respectively. He is currently pursuing a Ph.D. degree at the same university. His research interests include (1) additive manufacturing, (2) robotics, and (3) transportation logistics.

Dr. Harry Pierson is an Assistant Professor in the Department of Industrial Engineering at the University of Arkansas and directs the AT&T Manufacturing Automation Laboratory. He received his PhD from The Ohio State University in 2012 and has over 20 years of combined academic and industrial experience. He also holds an M.S. in Engineering Management and a B.S. in Mechanical Engineering from Missouri S&T. He studies collaborative robotics, additive manufacturing, and advanced manufacturing, and his research has been supported by the Army Research Office, Air Force Research Laboratory, and the Army Material Command.

Dr. Sarah Nurre Pinkley is an Assistant Professor in the Department of Industrial Engineering at the University of Arkansas. She received her PhD, MS, and BS from Rensselaer Polytechnic Institute. Her research interests include using network optimization, scheduling, and optimization algorithms for restoring interdependent infrastructure systems, operating electric vehicle and drone battery swap stations, understanding last-mile delivery, and optimizing complex systems.

Dr. Kelly M. Sullivan is an associate professor of Industrial Engineering at the University of Arkansas, Fayetteville, AR. His research focuses on advancing computational methodology for designing, maintaining, and securing complex systems. He holds a Ph.D. in Industrial and Systems Engineering from the University of Florida and a M.S. in Industrial Engineering from the University of Arkansas. Dr. Sullivan received a National Science Foundation CAREER Award in 2018 and was awarded the 2014 Glover-Klingman Prize for the best paper published in *Networks*. He is currently a member of IISE and INFORMS and serves as an associate editor for *Operations Research Letters* and *INFORMS Journal on Computing*.

References

- Antonio, F. (1992) IV.6 - Faster line segment intersection, in *Graphics Gems III* (IBM Version), David Kirk, San Francisco, pp. 199-202.
- Atwell, C. (2016) Autodesk's project escher uses multiple 3D print heads for massive jobs. Available at: <https://makezine.com/2016/03/24/autodesk-introduces-mass-3d-printing-project-escher/>.
- Bui, H., Pierson, H. A., Nurre, S. G. and Sullivan K. M. (2019) Tool path planning optimization for multi-tool additive manufacturing, *Procedia Manufacturing*, **39**, 457-464.
- Bui, H. (2019). Toolpath planning methodology for multi-gantry fused filament fabrication 3d printing, *Theses and Dissertations*, University of Arkansas, Fayetteville, Arkansas.
- Castelino, K., D'Souza, R., & Wright, P. K. (2003). Toolpath optimization for minimizing airtime during machining. *Journal of Manufacturing Systems*, **22**(3), 173–180.
- Chen, Z. C., Vickers, G. W., & Dong, Z. (2004). A New Principle of CNC Tool Path Planning for Three-Axis Sculptured Part Machining—A Steepest-Ascending Tool Path. *Journal of Manufacturing Science and Engineering*, **126**(3), 515–523.
- Choi, S. H. and Zhu, W. K. (2010) A dynamic priority-based approach to concurrent toolpath planning for multi-material layered manufacturing. *Computer-Aided Design*, **42**(12), 1095-1107.
- Compton, B. G. and Lewis, J. A. (2014) 3D-Printing of Lightweight Cellular Composites. *Advanced Materials*, **26**(34), 5930-5935.
- Cozier, A. D., Harned, K.E., Riley, M. A., Raabe, B. H., Sommers, A. D. and Pierson, H. A. (2015) Additive Manufacturing in the Design of an Engine Air Particle Separator. *Proceedings of the ASME 2015 International Mechanical Engineering Congress & Exposition*, Houston, TX.
- Fok, K. Y., Cheng, C. T., Ganganath, N., Iu, H. H. and Tse, C. K. (2018) Accelerating 3D Printing Process Using an Extended Ant Colony Optimization Algorithm. *Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, pp. 1-5.
- Ganganath, N., Cheng, C. T., Fok, K. Y. and Tse, C. K. (2016) Trajectory planning for 3D printing: A revisit to traveling salesman problem, in *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, Hong Kong, pp. 287-290.
- Glover, F. (2001) Tabu search, in *Encyclopedia of Operations Research and Management Science*, Springer US, Boston, MA, pp. 821-827.
- Greene, N. (2014) The 3&Dbot robotic 3D printer takes 3D printing on the road. Available at: <https://www.thecoolist.com/3dbot-robotic-3d-printer/>.
- Hmeidat, N. S., Kemp, J. W. and Compton, B. G. (2018) High-strength epoxy nanocomposites for 3D printing. *Composites Science and Technology*, **160**, 9-20.

- Jin, Y., Pierson, H. A. and Liao, H. (2017) Concurrent fused filament fabrication with multiple extruders, in *67th Annual Conference and Expo of the Institute of Industrial Engineers 2017*, Pittsburgh, pp. 940-945.
- Jin, Y., Pierson, H. A. and Liao, H. (2019) Toolpath allocation and scheduling for concurrent fused filament fabrication with multiple extruders. *IIE Transactions*, **51**(2), 192-208.
- Koenig, O. and Jouaneh, M. (2005) Minimization of airtime in cutting and welding applications. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 3300-3305.
- Kumar, R. (2012) Wing - 23012 aerofoil section. Available at: https://grabcad.com/library/wing-23012-aerofoil-section#_=_.
- LaValle, S. M. and Hutchinson, S. A. (1998) Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, **14**(6), 912-925.
- Lee, J. Y., An, J. and Chua, C. K. (2017) Fundamentals and applications of 3D printing for novel materials. *Applied Materials Today*, **7**, 120-133.
- Lee, K. H. and Kim, J. H. (2006) Multi-robot cooperation-based mobile printer system. *Robotics and Autonomous Systems*, **54**(3), 193-204.
- Li, N., Li, Y. and Liu, S. (2016) Rapid prototyping of continuous carbon fiber reinforced polylactic acid composites by 3D printing. *Journal of Materials Processing Technology*, **238**, 218-225.
- Li T. Y. and Latombe, J. C. (1995) On-line manipulation planning for two robot arms in a dynamic environment. *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan, pp. 1048-1055.
- Lin, Z., Fu, J., Shen, H., Gan, W. and Yue, S. (2015). Tool path generation for multi-axis freeform surface finishing with the LKH TSP solver. *Computer-Aided Design*, **69**, 51–61.
- Marques, L. G., Williams, R. A. and Zhou, W. (2017) A mobile 3D printer for cooperative 3D printing, *Proceeding of the 28th International Solid Freeform Fabrication Symposium*, Texas, pp. 1645-1660.
- Netfabb (2018) Autodesk Knowledge Network. Available at Autodesk website: <https://knowledge.autodesk.com/support/netfabb/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/NETF/files/GUID-E2E1110F-CA7F-4580-8E9C-250B796B6E05-htm.html>
- Oysu, C. and Bingul, Z. (2007) Tool path optimization using genetic algorithms. *Proceedings of the 2007 international conference on genetic and evolutionary methods (GEM)*, Las Vegas, NV, pp. 120–126.
- Pierson, H. A., Celik, E., Abbott, A., DeJarnette, H., Gutierrez, L. S., Johnson, K., Koerner, H. and Baur, J. W. (2019) Mechanical properties of printed epoxy-carbon fiber composites, *Experimental Mechanics*, **59**, 843-857.

- Peasgood, M., Clark, C. M. and McPhee, J. (2008) A complete and scalable strategy for coordinating multiple robots within roadmaps. *IEEE Transactions on Robotics*, **24**(2), 283-292.
- PLM Technology (2019) Bracket topology optimization 2. Available at: <https://grabcad.com/library/bracket-topology-optimization-2-1>.
- Rodrigues, A. M. and Ferreira, J. S. (2001). Solving the rural postman problem by memetic algorithms. *Proceedings of the 4th Metaheuristic International Conference (MIC'2001)*, Porto, Portugal, pp. 679-684.
- Salaets, B. (2018) Rotor hub wind turbine. Available at <https://grabcad.com/library/rotor-hub-wind-turbine-1>.
- Saunders, Marc. (2018) Proximity pays - how multiple lasers can work together on high-integrity parts. Available at: <https://www.linkedin.com/pulse/proximity-pays-how-multiple-lasers-can-work-parts-marc-saunders/>.
- Titan Robotics (2017) The Cronus. Available at: <https://titan3drobotics.com/the-cronus/>.
- Todt, E., Rausch, G. and Suarez, R. (2000) Analysis and classification of multiple robot coordination methods. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 3158-3163.
- Volpato, N., Nakashima, R. T., Galvao, L. C., Barboza, A. O., Benevides, P. F. and Nunes, L. F. (2014) Reducing repositioning distances in fused deposition-based processes using optimization algorithms, in *6th International Conference on Advanced Research in Virtual and Rapid Prototyping*, Leiria, Portugal, pp. 417-422.
- Wagner, I. A., Altshuler, Y., Yanovsky, V. and Bruckstein, A. M. (2008) Cooperative cleaners: a study in ant robotics. *The International Journal of Robotics Research*, **27**(1), 127-151.
- Wah, P. K., Murty, K. G., Joneja, A. and Chiu, L.C. (2002) Tool path optimization in layered manufacturing. *IIE Transactions*, **34**(4), 335-347.
- Williams, K. and Burdick, J. (2006). Multi-robot boundary coverage with plan revision. *Proceedings 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, pp. 1716-1723.
- Wojcik, M., Koszalka, L., Pozniak-Koszalka, I. and Kasprzak, A. (2015). MZZ-GA algorithm for solving path optimization in 3D printing. *Proceedings of the Tenth International Conference on Systems (ICONS 2015)*, pp. 30-35.
- Zhang, X., Li, M., Hui Lim, J., 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**, 98-106.

Figure 1: a) Illustration the OSA method; b) output of the proposed method with automatically-generated segmentation that preserves infill pattern.

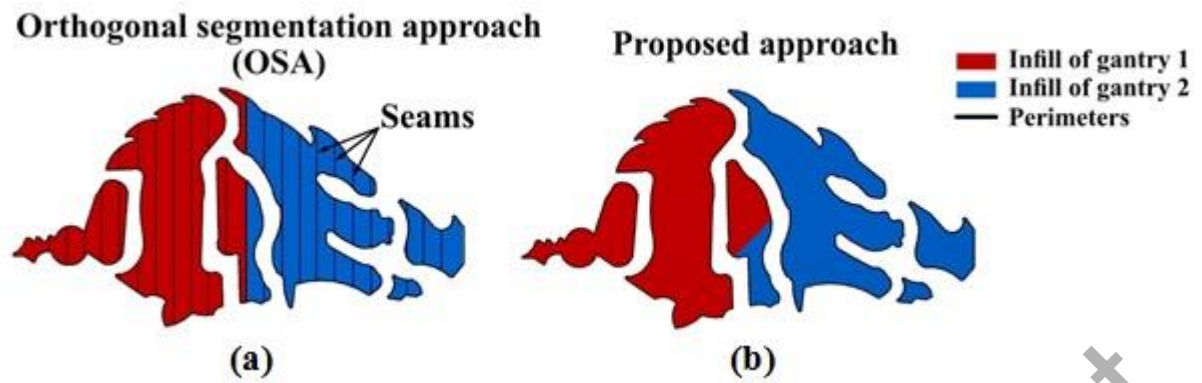


Figure 2: Raster segments with identification numbers generated from an IE Hog layer (2% infill was selected for demonstration purposes).

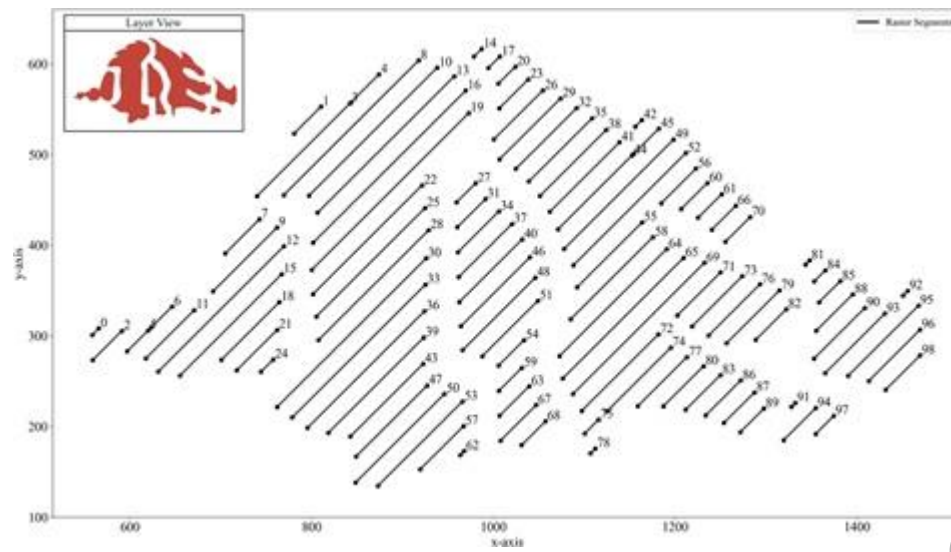


Figure 3: Illustration of the GS, LS, and rebalancing operators.

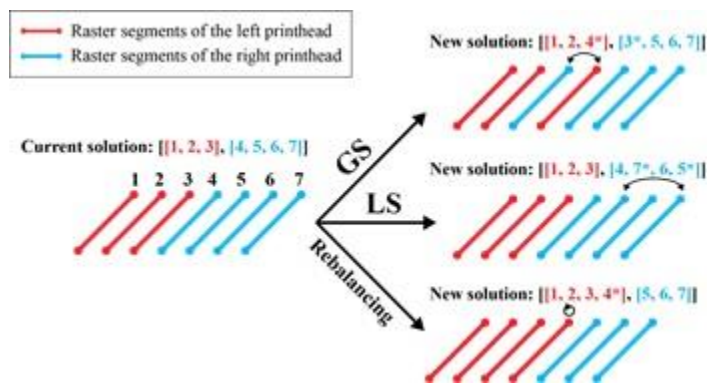


Figure 4: Flowchart of the tabu search algorithm.

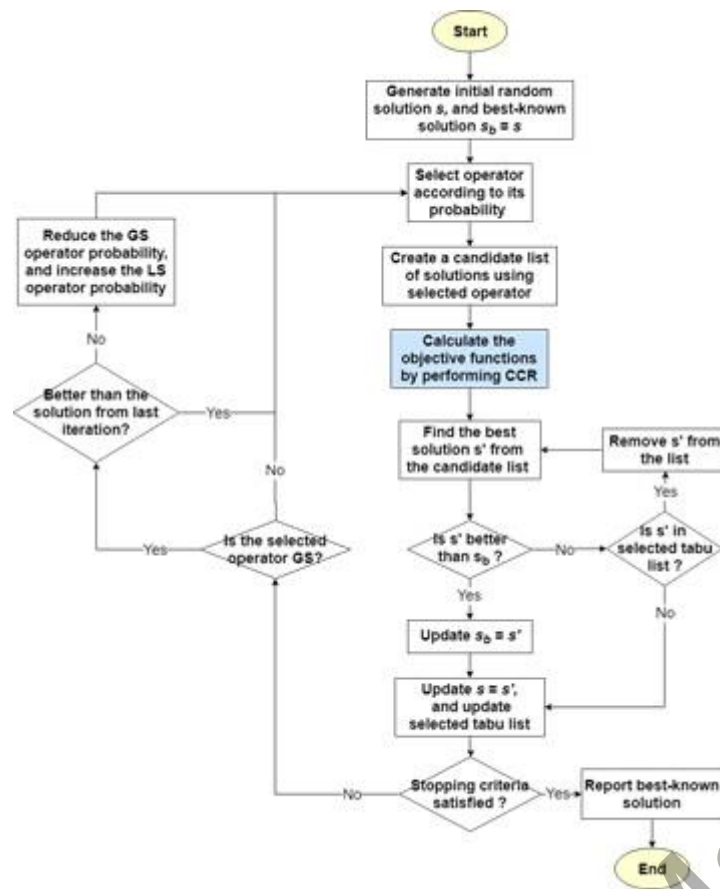
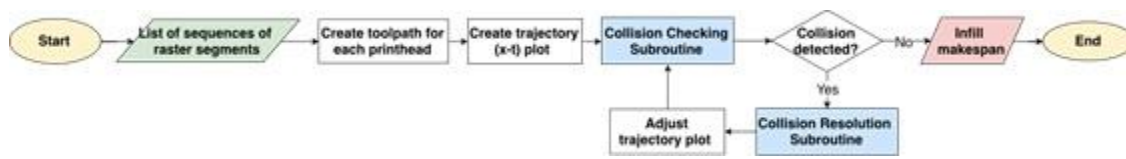


Figure 5: Flowchart of the solution evaluation with the help of CCR.



Accepted Manuscript

Figure 6. Toolpath representation of a potential solution.

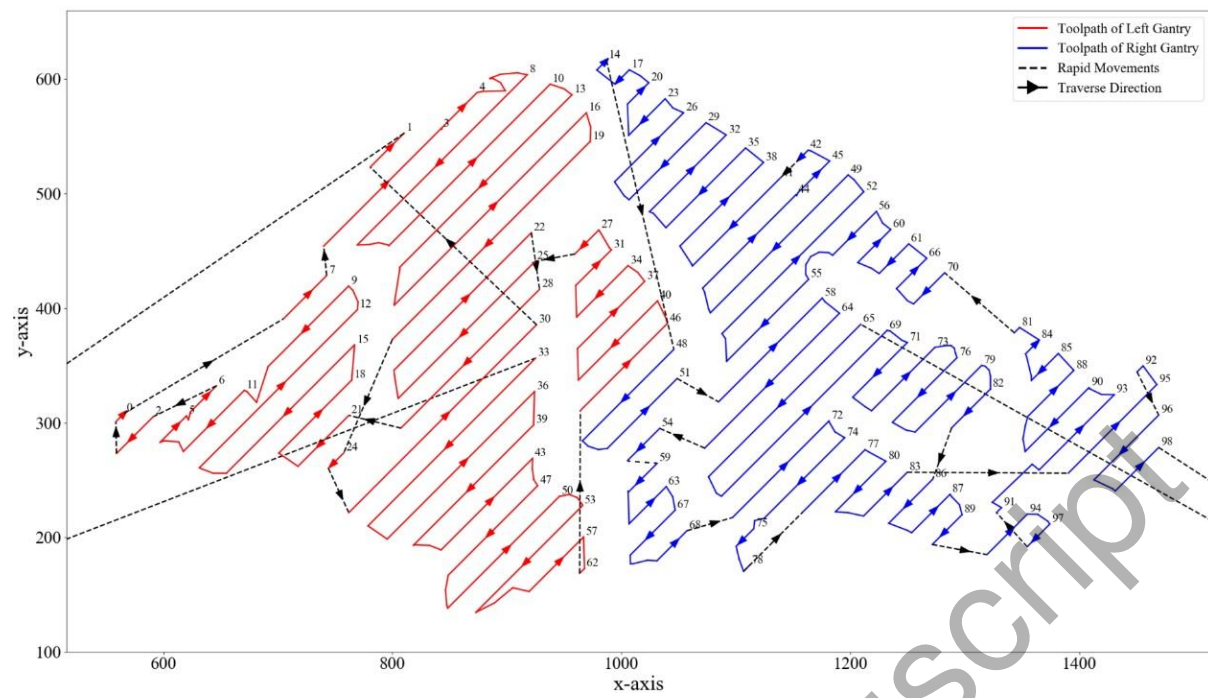


Figure 7. Illustration of the calculation of the x-coordinate of each gantry.

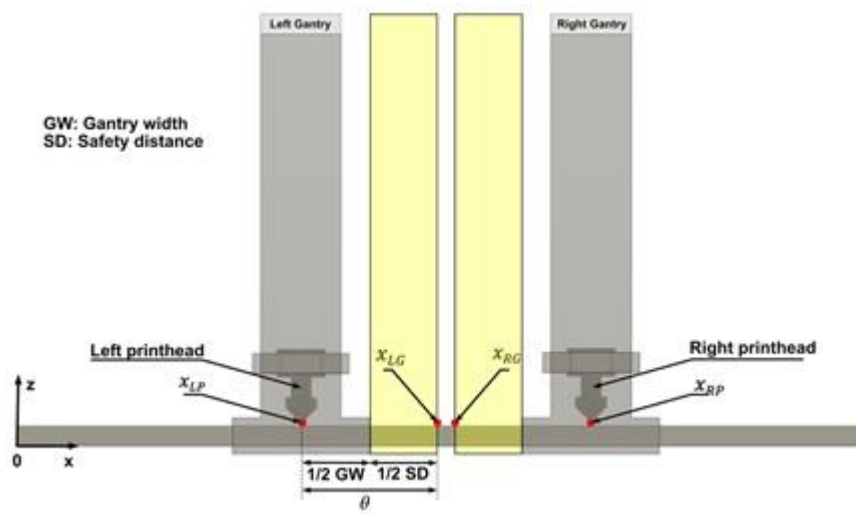


Figure 8. Trajectory plot of the toolpaths in Figure 6, collision checking subroutine identified 6 potential collisions.

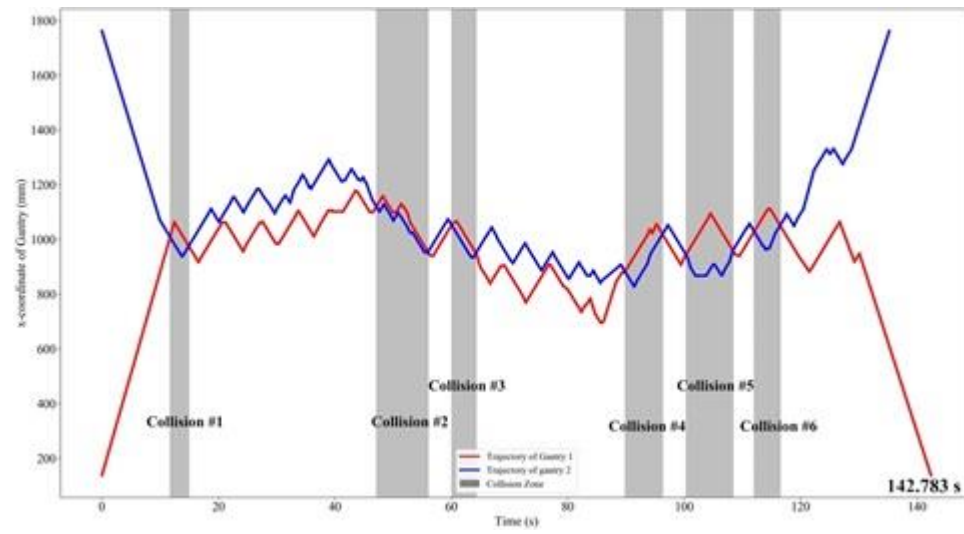


Figure 9. Final trajectory plot result from the CCR. Several delays were added to solve the 6 collisions.

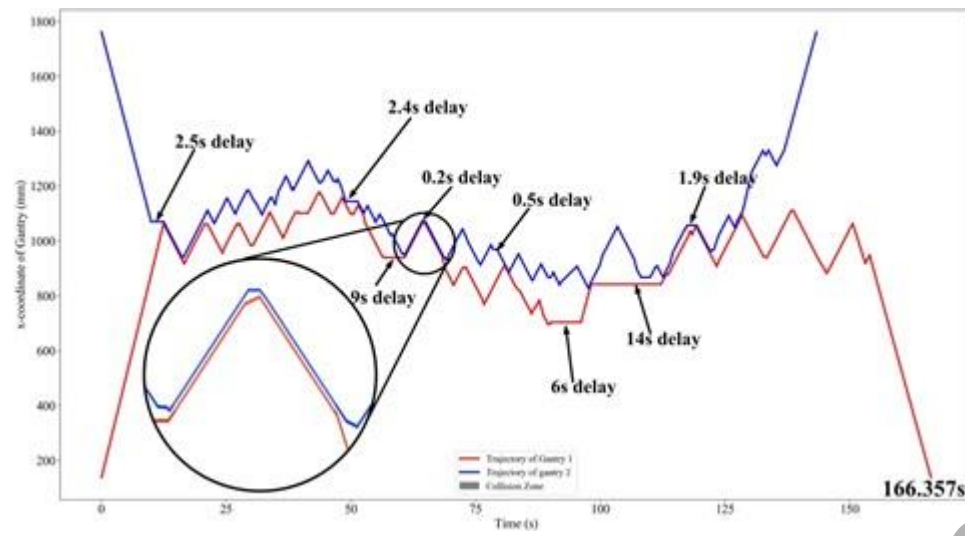


Figure 10. a) calculations to determine the value of d^l ; b) calculations to determine the value of d^r ; c) collision resolution subroutine added the delay to the left trajectory because it required less delay to resolve the collision.

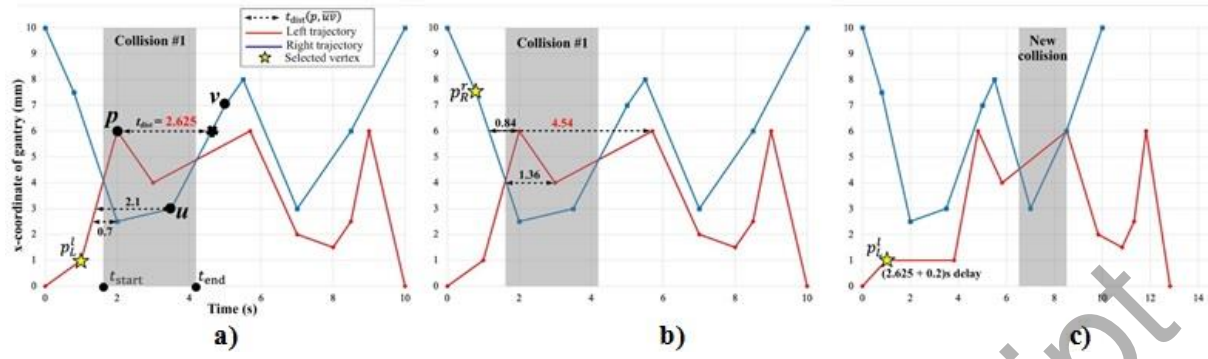


Figure 11. Flowchart of closed-loop control with the resynchronization process built-in for two-gantry configuration.

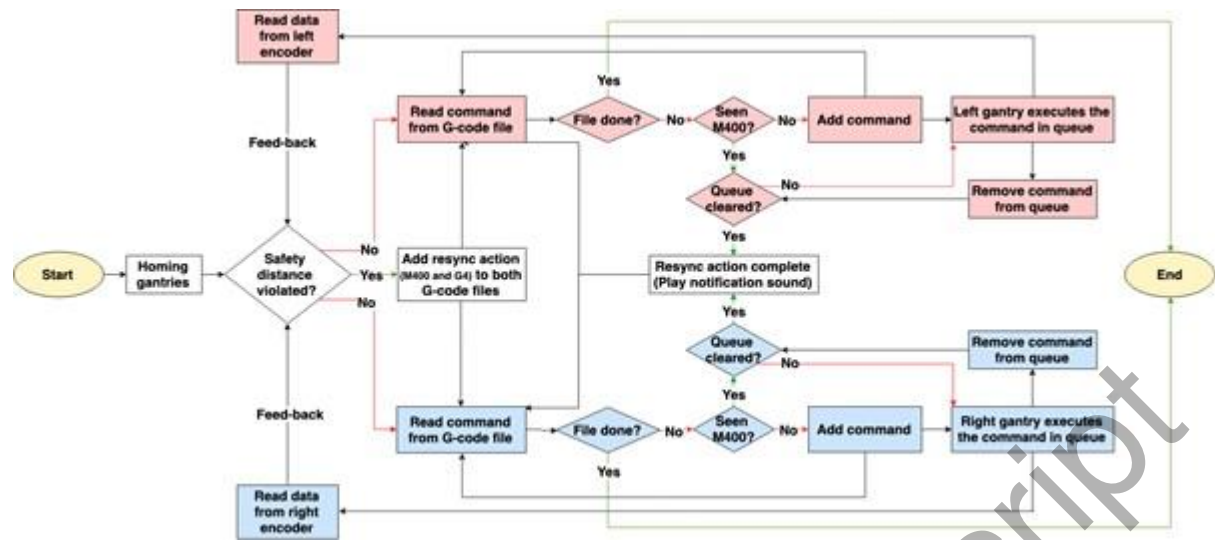


Figure 12. a) Layer from IE Hog; b) layer from airfoil frame (Kumar, 2012); c) layer from topology optimized bracket (PLM Technology, 2019); d) layer from rotor hub wind turbine nose (Salaets, 2018).

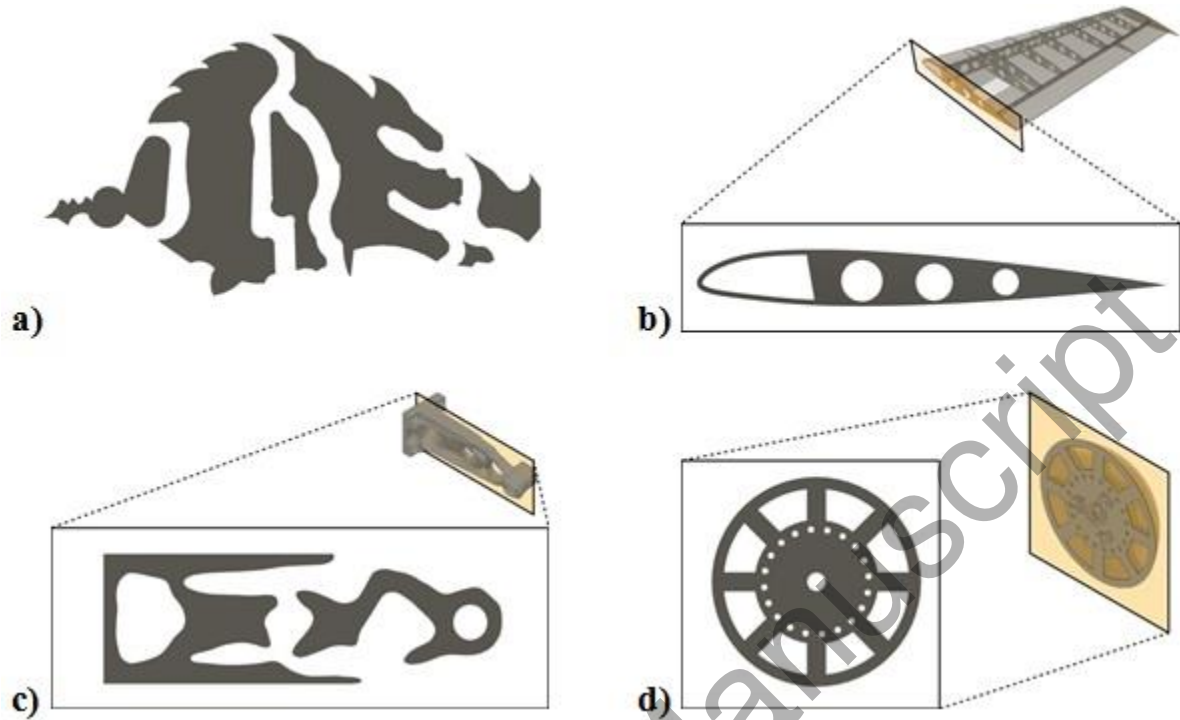


Figure 13. a) The custom two-gantry 3D printer; b) a custom bracket that holds x-axis motor and rotary encoder.

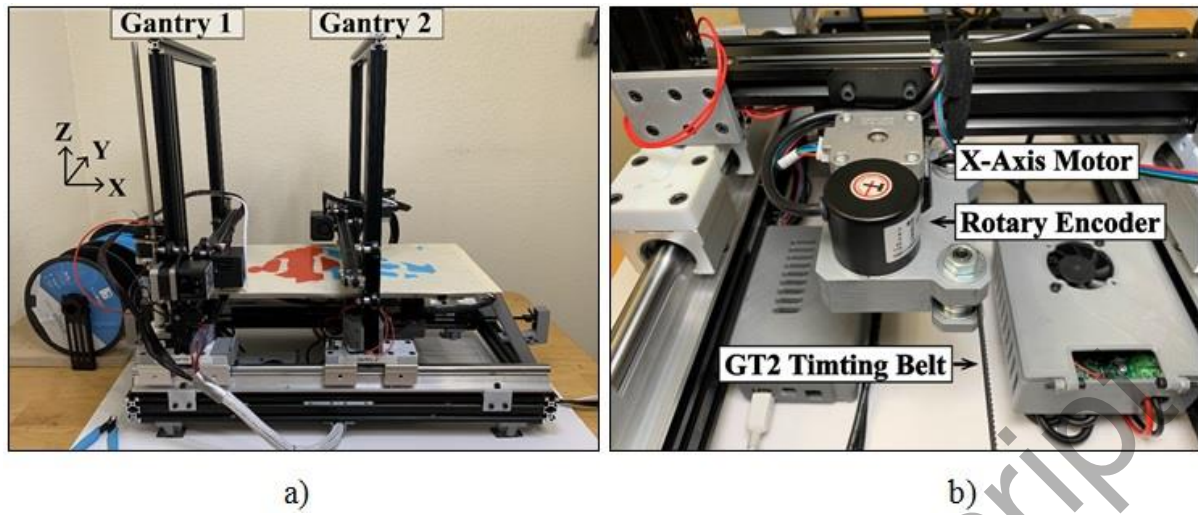


Figure 14. Closeup shot of the seam.



Accepted Manuscript

Figure 15. Makespan comparison of four selected layers at 30% infill, where the proposed TS-CCR can yield solutions with shorter makespan than the solution obtained from OSA.

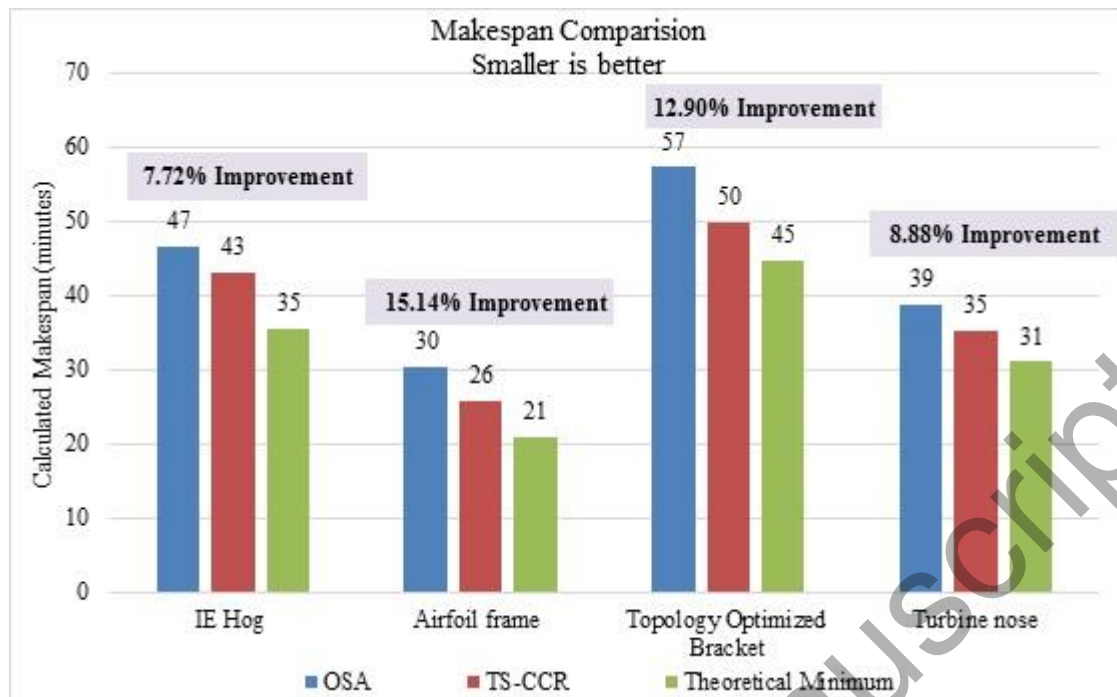


Figure 16. Failures of specimens printed by two different methods

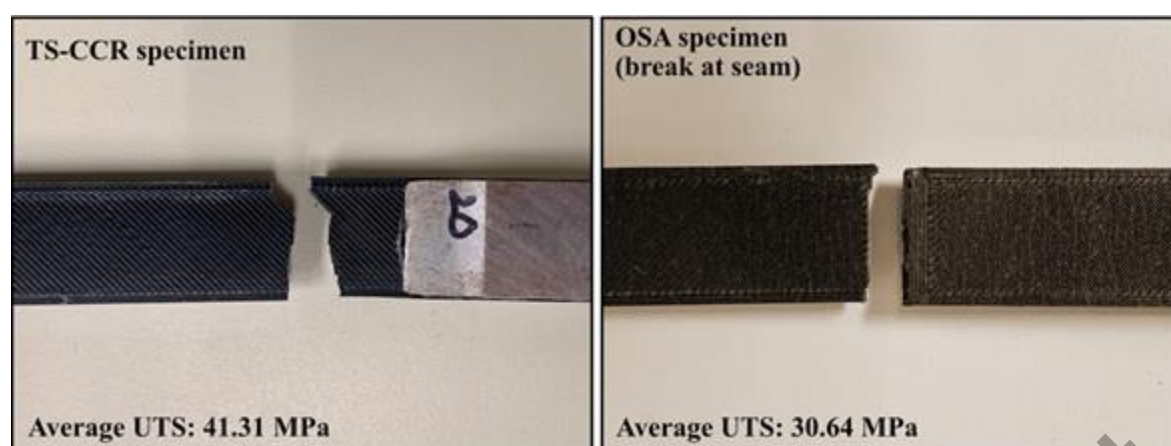


Figure 17 a) The custom setup consists of two independent gantries, each printing the assigned part. While printing, the gantries never touched each other; b) closeup of the complete layer where two infill portions connect.

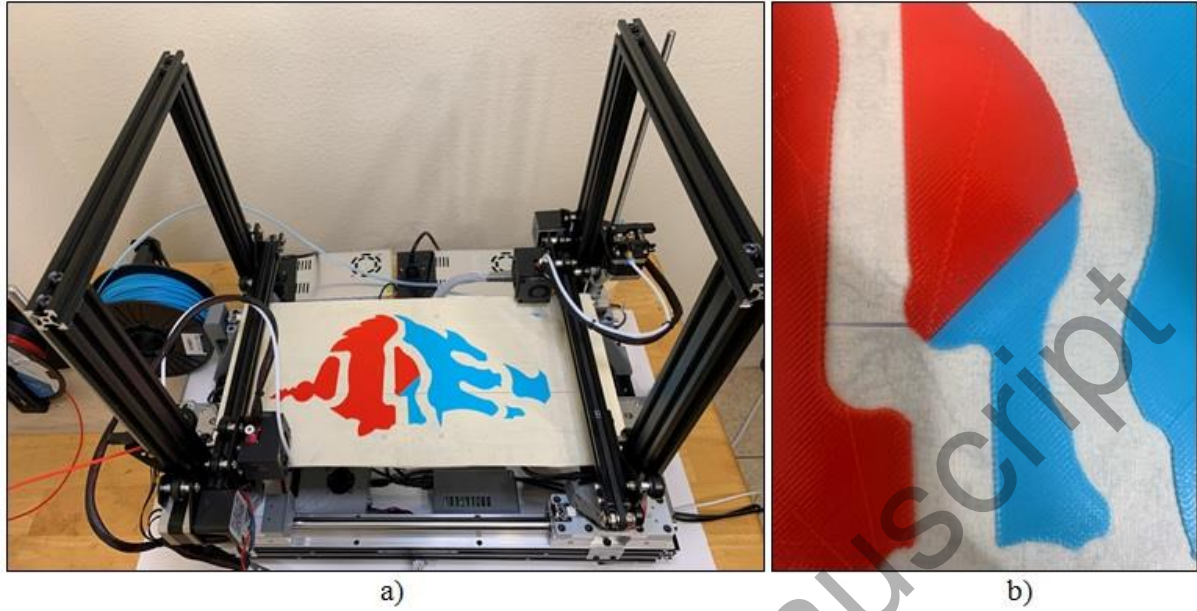


Figure 18 a) Trajectory plot generated from the TS-CCR for “IE Hog” (zoomed section shows that the two trajectories do not overlap); b) trajectory plot generated from the encoders.

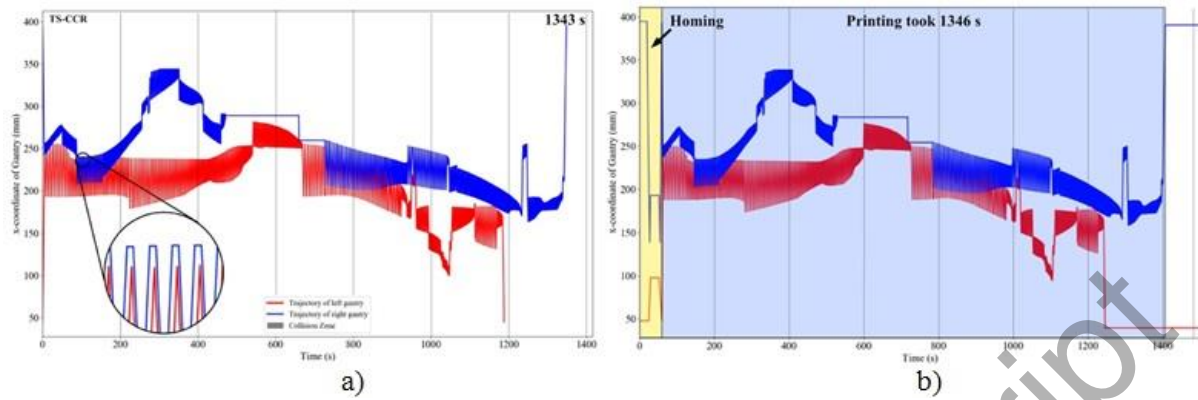


Table 1. Printing parameters for 3D printed parts

Nozzle diameter	0.4 mm
Layer thickness	0.30 mm
Number of perimeter shells	3
Default printing speed	50 mm/s
Raster angle offsets	45 and -45 degrees
Infill percentage	100%
Extrusion temperature	210°C

Table 2. Results from tensile testing.

	UTS (MPa)					Average	STD
OSA	33.91	34.80	35.32	24.02	25.13	30.64	5.57
TS-CCR	40.93	41.68	41.36	41.51	41.08	41.31	0.31