Delft University of Technology

Material Deposition in 3D Space
Additive Manufacturing Enriched by Rotational Motion

Dai, C.

DOI
10.4233/uuid:60ea0ffa-ef45-4995-af17-f618c73ea519

Publication date
2020

Document Version
Final published version

Citation (APA)
Dai, C. (2020). *Material Deposition in 3D Space: Additive Manufacturing Enriched by Rotational Motion*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:60ea0ffa-ef45-4995-af17-f618c73ea519

Material Deposition in 3D Space

Additive Manufacturing Enriched by Rotational Motion

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Thursday, 16 July 2020 at 15:00 o'clock

by

Chengkai DAI

Master of Science in Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Hong Kong SAR, China,
born in Wuxi, China.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof.dr.ir. C.C.L. Wang	Delft University of Technology, promotor
Prof.dr.ir. J.M.P. Geraedts	Delft University of Technology, promotor

Independent members:

Prof.dr. E. Eisemann	Delft University of Technology
Prof.dr. S.C. Pont	Delft University of Technology
Dr. Y. Song	Delft University of Technology
Dr. O.W. Salomons	Canon Production Printing Netherlands B.V.
Prof.dr.ir. K.M.B. Jansen	Delft University of Technology, reserve member

Other member:

Dr. S. Lefebvre	INRIA, France

Keywords: Additive Manufacturing, 3D Printing, Robotics

Printed by: Ipskamp Printing, the Netherlands

Front & Back: Explanation

ISBN 978-94-028-2096-6

Dedicated to my family.

Table of Contents

Summary

Additive manufacturing (AM) is causing a revolution in product design, manufacturing and distribution. This technology facilitates the production of complex, customized products without the need of any specific tooling, thereby enabling products to be delivered at a lower cost than with traditional manufacturing. From a design perspective, AM allows designers to selectively place (multi-)material where it is needed to achieve the designed functionality. However, despite remarkable progress in the domain of AM, a variety of challenges – like support structures, staircase effects, and mechanical performance [1] – should be investigated at depth to fully explore the potential of AM. On the other hand, these challenges limit the designers' freedom to realize their creativity.

One of the reasons that cause of the challenges listed above in the conventional AM system is a lack of orientation change. Adding rotational motion to an AM system can significantly increase the flexibility of material deposition (from planar into 3D), thereby allowing the possibility of eliminating support structures, improving surface quality and enhancing mechanical performance. To prove this concept, a novel fused-deposition-modeling (FDM) based robot-assisted AM setup with dedicated algorithms has been developed.

The major contribution of this thesis is workflow development of the robot-assisted AM system to overcome the limitations of conventional AM technology. The same strategy of conventional AM workflow is used, but it has been adjusted in this thesis to be applied in the 3D domain. This conventional workflow consists of three steps: slicing, tool-path generation, and hardware realization.

For the slicing part, complexity explodes with the change from the planar to the 3D domain. This thesis presents a new way to tackle this challenge by optimizing a scalar field within the volume that represents the fabrication sequence. The 3D model is first discretized into a regular voxel grid. Then the material accumulation during AM is assumed to be performed by adding voxels one by one. Several algorithms are proposed to search for an accumulation sequence by taking into consideration the manufacturing needs, hardware constraints, and computation efficiency. This sequence is encoded by storing an integer (rank in sequence) at the center of each voxel to represent the final scalar field. In the end, the sliced layers are extracted from this scalar field as isosurfaces at different isovalues.

Unlike tool-path generation in CNC milling, which focuses primarily on forming the surface of a part by cutting material, robot-assisted AM faces the problem of filling the volume of the part with evenly spaced, non-overlapping trajectories, which are always superimposed on an already solidified volume (no isolated, floating components possible). In this thesis, a special space-filling pattern, called Connected Fermat spirals (CFS), is extended to the curved surface domain. The method developed in this study for tool-path generation both meets the requirements of uniform space and non-overlap and offers the advantage of position continuity and orientation continuity, thereby resulting in continuous material deposition and the smoothest possible robot movement.

On the hardware side, during the fabrication, the robotic arm follows prescribed tool-paths generated by the above-mentioned method to fill the sliced surfaces in cooperation with the extrusion head. The robot movement must be optimized due to the non-linear mapping between the waypoint representation in tool-paths and robot's configuration space. The kinematic redundancy that existed in the system also should be considered. To improve motion smoothness and fabrication quality, this thesis proposes a novel jerk-minimized motion planning method for redundant robots. The method is based on a sampling strategy. After determining an initial path by graph search, a greedy algorithm is adopted to optimize a path by locally applying adaptive filters in the regions with large jerks. To achieve efficient computation, an adaptive sampling method is developed for learning a collision-indication function that is represented as a support-vector machine.

Finally, a new trend of AM process that facilitates fabricating the parts with improved mechanical properties is explored by combining the FDM-based robot-assisted AM system with continuous-fiber material. This new AM approach reinforces the mechanical performance by two steps. First, the robotic system allows filaments to be oriented according to the stress distribution based on the loading direction, thereby achieving the optimal arrangements for the thermoplastic strands; second, continuous carbon-fibers are directly deposited on the printing part in a sandwich structure. This fiber-deposition method further improves the mechanical performance and speeds up the whole process because of the much wider fiber width. Preliminary physical tests have been conducted for this thesis and notable structure enhancement has been observed in the experimental results.

As one of the reviewers of our SIGGRAPH paper said, “I think the paper makes an important step towards using the full capabilities of higher-DOF 3D printing and opens the door to an interesting new fabrication approach.”.

Hopefully, the research presented in this thesis will have an impact on the additive manufacturing community, lead to better and more reliable products, and inspire future research.

Samenvatting

Additive manufacturing (AM) brengt een revolutie teweeg in de manier waarop producten worden ontworpen, gefabriceerd en gedistribueerd. De technologie faciliteert de productie van complexe, op maat gemaakte producten zonder dat hiervoor onderdeel-specifiek gereedschap nodig is, en maakt het daarmee mogelijk producten tegen een lagere prijs te leveren dan met traditionele fabricatie technieken mogelijk zou zijn. Vanuit een ontwerp-perspectief biedt AM ontwerpers de mogelijkheid selectief (een combinatie van meerdere) materialen te plaatsen, precies daar waar het nodig is om een ontworpen functionaliteit te realiseren. Ondanks opzienbarende vooruitgang in het AM-domein zijn er een aantal uitdagingen - zoals in de ondersteunende structuren, de trap-effecten, en de mechanische eigenschappen [1] - die aandacht nodig hebben om de volle potentie van AM te kunnen waarmaken. In andere woorden: deze uitdagingen beperken de vrijheid van de ontwerper in het realiseren van hun creativiteit.

Een van de redenen die ten grondslag ligt aan de hierboven genoemde uitdagingen is het ontbreken van veranderingen in oriëntatie. Het toevoegen van een rotationele beweging aan AM-systemen, kan de flexibiliteit van de materiaalopeenstapeling significant vergroten (van planair naar 3D), waardoor het mogelijk wordt ondersteunende structuren te elimineren, de oppervlaktekwaliteit te verhogen en de mechanische eigenschappen te verbeteren. Om dit concept te bewijzen is een nieuwe FDM-gebaseerde, robot-geassisteerde AM-opstelling met benodigde algoritmes ontwikkeld.

De hoofdbijdrage van deze thesis is de optimalisatie en volgorde van machine instellingen (workflow) om de beperkingen van conventionele AM-technologieën te overwinnen. Hierin wordt dezelfde strategie als in conventionele AM geadopteerd, maar zijn deze stappen opnieuw ontwikkeld om ze toe te kunnen passen in het driedimensionale (3D) domein. De conventionele strategie bestaat uit drie stappen: ‘slicing’, de generatie van de machine instellingen, en de realisatie van de hardware.

De complexiteit van de slicing neemt aanzienlijk toe wanneer het domein aangepast wordt van planair naar 3D. In deze thesis wordt een nieuwe methodologie gepresenteerd om deze uitdaging aan te pakken door het optimaliseren van een scalair veld binnen het volume dat de fabricage-volgorde representeert. Het 3D-model wordt eerst gediscretiseerd in een regelmatig voxel-raster. Vervolgens wordt aangenomen dat de materiaal opeenstapeling tijdens AM wordt uitgevoerd door het één voor één plaatsen van voxels.

Verschillende algoritmes zijn geïntroduceerd voor het zoeken naar een fabricatie volgorde, waarbij rekening gehouden wordt met de fabricatie eisen, beperkingen van de hardware en de computationele efficiëntie. De volgorde wordt gecodeerd door een geheel getal (rang in volgorde) in het centrum van elke voxel op te slaan die het uiteindelijke scalaire veld representeert. Uiteindelijk worden de lagen uit dit scalaire veld geëxtraheerd als iso-oppervlakken met verschillende iso-waarden.

Bij CNC-frezen richt de generatie van de workflow zich op het vormen van het oppervlak van een onderdelen door het wegsnijden van materiaal. Robot-geassisteerde AM loopt tegen het probleem aan van het vullen van het volume van het onderdelen met gelijkmatig verdeelde, niet overlappende paden, die altijd neergelegd worden bovenop eerder uitgeharde volume-elementen (ter voorkoming van geïsoleerde 'zwevende' componenten). In deze thesis is een speciaal ruimte-vullend patroon gebruikt, dat "Connected Fermat Spirals (CFS)" genoemd wordt, en uitgebreid naar het domein van gekromde oppervlakten. De workflow generatie methode die in dit onderzoek ontwikkeld is voldoet aan de eisen van gelijkmatige verdeling en absentie van overlap en biedt ook het voordeel van positie-continuïteit en oriëntatie-continuïteit, wat leidt tot continue materiaal depositie en geleidelijke robotbewegingen.

Bij het fabriceren van het onderdelen volgt de robot een voorgeschreven beweging, welke gegenereerd wordt door de hierboven genoemde methode te gebruiken om de gelaagde oppervlakte met de extrusiekop te vullen. De beweging van de robot moet geoptimaliseerd worden, aangezien vertaling tussen de cartesische ruimte en de robot configuratie ruimte niet-lineair is. De kinematische overbepaaldheid in het systeem moet hierin ook meegenomen worden. In deze thesis wordt een nieuwe robotarm beweging geïntroduceerd die het aantal schokken minimaliseert om de beweging geleidelijker te maken en daarmee betere fabricatie kwaliteit mogelijk te maken. De methode is gebaseerd op een sampling-strategie en bestaat uit twee hoofdcomponenten. Na het bepalen van het initiële pad door een zoekalgoritme voor grafen, wordt een inhalig algoritme gebruikt om een pad te optimaliseren door lokaal adaptieve filters toe te passen in gebieden met grote schokken. Om dit efficiënt te kunnen berekenen, is een adaptieve sampling methode ontwikkeld voor het leren van een botsingsindicatie functie die gerepresenteerd wordt als een support-vector machine.

Tot slot wordt een nieuwe trend verkend die het mogelijk maakt onderdelen met verbeterde mechanische eigenschappen te fabriceren door het FDM-gebaseerde, robot-geassisteerde AM-systeem te combineren met doorlopende koolstof draden (fibers). Deze nieuwe AM-aanpak verhoogt de mechanische eigenschappen in twee stappen. Allereerst maakt het robotische systeem het mogelijk de thermoplastische draad te oriënteren in de richting van de belasting en daarmee een optimaal arrangement van de thermoplastische draden te realiseren. In de tweede stap worden doorlopende koolstoffibers direct op het geprinte onderdeel geplaatst in een sandwich structuur. Deze manier van materiaal opeenstapeling verbetert niet alleen de mechanische eigenschappen, maar versnelt ook het fabricageproces omdat de fibers een grotere doorsnede hebben. In dit proefschrift worden inleidende metingen beschreven die al een significante verbetering van de structuur opleveren.

Zoals een van de reviewers van de SIGGRAPH-publicatie zei, “Ik denk dat de publicatie een belangrijke stap is richting het volledig benutten van de mogelijkheden van 3D-printen met meer vrijheidsgraden en het een deur opent naar een interessante nieuwe benadering voor fabricatie”. Ik hoop dat het werk dat in deze thesis wordt gepresenteerd zijn impact zal hebben op de AM-gemeenschap, zal leiden tot betere en betrouwbaardere producten, en toekomstig werk zal inspireren.

Glossary

3D The domain of rigid transformation in three dimensions.

3D Printing The same meaning as *Additive Manufacturing* in this thesis.

Additive Manufacturing A manufacturing method that building three dimensional objects from digital files, usually by adding material in a layer-wise manner.

Anisotropic Properties The properties that vary with respect to direction.

CAD Computer-aided Design, is the use of computers to aid the creation of digital model.

CAM Computer-aided Manufacturing, is the use of computers to create the programs that will control machine movement in manufacturing process.

CNC Computer Numerical Control, is the use of computers to control the machinery used in manufacturing process.

Composite Material A material produced from two or more constituent materials with different mechanical performance.

Configuration Space The space of all (joint) positions the robot may attain.

Curved Surfaces The locus of all points of a moving and deforming curve.

Degree of Freedom The number of independent displacements or motion that affecting the movement of a system.

Euclidean Distance The length of a direct segment connecting the two points in Euclidean space.

FDM Fused Deposition Modeling, is an additive manufacturing process that uses a continuous filament of a thermoplastic material.

Fermat Spiral A special space-filling pattern with two interleaving sub-spirals, one inward and one outward [2].

Freeform Surfaces The same meaning as *Curved Surfaces* in this thesis.

Genus The number of “holes” of a surface.

G-Code The machine language generated from tool-paths that instructs machines where and how to move.

IK Inverse kinematics, is the mathematical process to calculate the joint displacements of a kinematic chain with a given position and orientation.

Jerk The rate of acceleration change with respect to time.

Kinematic Redundancy occurs when robot has more degrees of freedom than required for a given task.

Mechanical Performance The performance that exhibits the reaction to the application of forces.

NURBS Non-uniform Rational Basis Spline, is a common way to represent curves and surfaces.

Orientation The same meaning as *Rotation* in this thesis.

Overhang The region in 3D model that sticks out or hangs over another thing.

Robotic Arm The type of robot consists of parts linked together, with similar functions to a human arm.

Robot-assisted AM The AM system with rotational motion provided by robotic arm.

ROS Robot Operation System, is robotics middleware for building robot applications.

Rotation The circular movement of an object around a center.

SIGGRAPH Special Interest Group on Computer GRAPHics and Interactive Techniques, is the world's largest, most influential annual conference on the theory and practice of computer graphics and interactive techniques.

Slicing One step in the AM process that slices a 3D model into a set of surfaces so that each one can be fabricated in sequence.

Support Structure The material which is used to hold up the overhang region in printed model to prevent collapse.

Support Vector Machine A machine learning algorithm which can be used for classification and regression analysis.

Staircase Effect AM uses parallel planes to approximate a curved surface, thereby resulting in non-smooth edge that looks like a staircase.

Tool-path A prescribed path that the extrusion nozzle follows on its way to fill the volume of the sliced surface.

Topological Optimization The method that optimizes material distribution according to a given set of loads, boundary conditions to achieve the most efficient design.

Translation The motion involves the sliding of an object in a line.

Voxel defines the data resolution, it is the smallest element in a three-dimensional array.

Waypoint An intermediate point on the tool-path. In robot-assisted additive manufacturing, it is represented by the position along with the direction of the extrusion nozzle.

1

Introduction

1.1 Introduction

Industrial designers develop concepts for manufactured products as diverse as cars, home appliances, and toys. On the other hand, manufacturing is the key process whereby industrial designers realize their creative concepts. A designed product comes to life mainly in three manufacturing methods: (1) A forming/deforming method, such as injection molding, forms or deforms the right amount of bulk material to the right shape; (2) a decremental method, such as milling, gradually removes all excess material from a larger amount of bulk material; [3] or (3) an incremental method, such as FDM 3D printing, gradually adds material until the required shape is created. The first two manufacturing methods are decades old and have accomplished a symphony with science and engineering. The resulting mature supply-chain management of these methods has allowed them to dominate the manufacturing market for more than a century, and they will continue to dominate it for a long time to come. The third method is also called additive manufacturing (AM), which is the youngest technology of the three, but its development has progressed the most recently.

Additive manufacturing, the heart of this thesis, was initially invented for rapid prototyping, and in the mean time it has made a significant contribution to the design process. Unlike deforming and decremental methods that require expensive machinery and high workloads, it can directly fabricate 3D models using desktop-level hardware without any tool, jig, or fixture. As a result, AM can facilitate design adaptations on the fly with multiple iterations in a short time without paying extra cost penalty, thereby accelerating product development and even contributing to better design. In addition, AM also allows designers to customize their designs to suit different personal needs, as shown in Figure.1.1. Additive manufacturing is not merely revolutionizing the product-design process, however; it is also transforming methods of manufacturing across multiple industries (e.g., automotive, aerospace, maritime, medical, space, sports, motorsports, railway, and defense) [4].

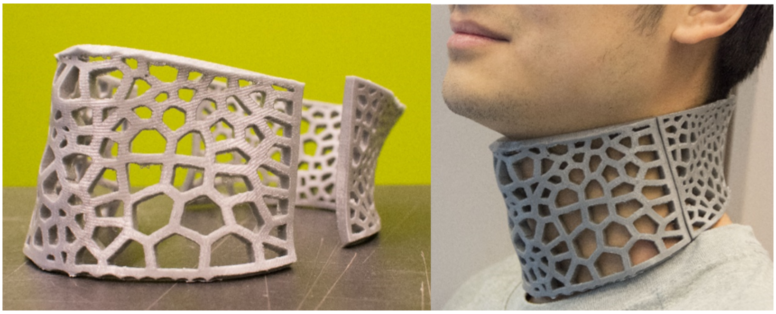


Figure 1.1: Courtesy of AM use: Customized 3D printed thermal-comfort neck cast for improving the rehabilitation of injured people [5].

Despite remarkable progress, AM still represented less than 0.1% of the total global manufacturing value in 2019 [6]. One reason is that the technology is still facing multiple issues that restrict its use in small-scale production and mass manufacturing. Some issues also prevent designers from designing any shape they can think of. To push the boundaries of current AM technology, this study was initiated to investigate the reasons behind the challenges that designers are facing while using AM (specifically, FDM) and to explore solutions with which to tackle these challenges.

The fundamental principle is briefly overviewed in the following section.

1.2 Principle of current AM technology

Additive manufacturing (AM), also known as 3D printing, comprises a group of technologies used to build physical parts by adding material in a layer-wise manner.

The standard terminology for AM has been developed by the ASTM (American Society for Testing and Material) [7]. Existing AM processes are classified into seven categories, including vat photopolymerization, powder-bed fusion, binder jetting, sheet lamination, directed energy deposition, material jetting, and material extrusion. A complete survey of all AM processes is beyond the scope of this thesis. Detailed reviews of state-of-the-art methods can be found in a number of papers and books on this topic. Two of these, [8, 9], are recommended to readers who are interested. As the material-extrusion method (specifically, FDM) is the most popular AM technique among designers [10], it is central to this thesis.

Due to its process simplicity, Fused Deposition Modeling (FDM), which was initially proposed by S. Scott Crump in the 1980s and commercialized by Stratasys, is the most commonly used material-extrusion process [10].

The FDM process mainly includes the following three steps.

1. Filament material, usually thermoplastic, is uncoiled from the spool into a heated extrusion nozzle, where it is heated and melted.
2. The extrusion nozzle follows the prescribed planar locations generated by CAM software (also called Slicer) in XY axes, while the melted material is extruded in thin strands, deposited, and solidified.
3. After each layer is deposited, the build-platform moves along the Z axis, increasing the distance between the nozzle and the build platform to deposit a new planar layer on top of the previous layer.

This process is repeated until the part is completed, as illustrated in Figure.1.2.

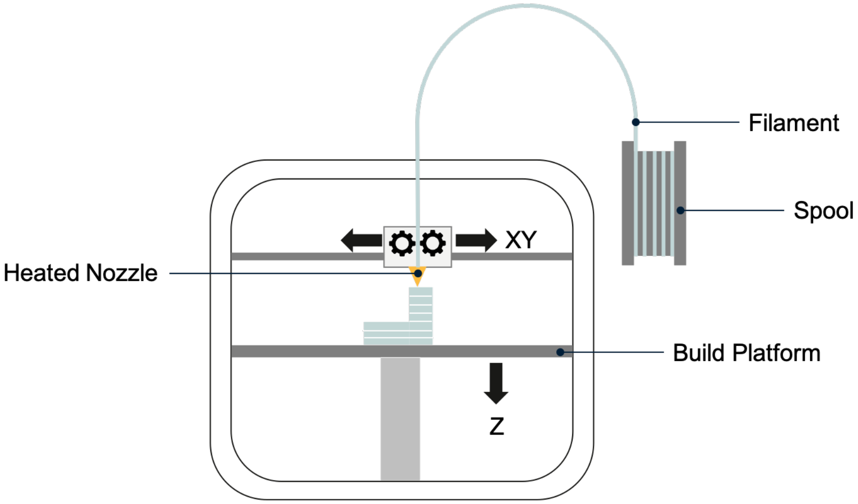


Figure 1.2: Illustration of a conventional FDM process.

The thermoplastic materials that are commonly used in FDM are acrylonitrile butadiene styrene (ABS), polylactic acid (PLA), thermoplastic polycarbonate (TPU), and nylon. Different materials lead to parts with different mechanical performance, so designers should carefully choose the right material before fabrication. Beside materials, many operational parameters – including layer height, nozzle diameter, extrusion feedrate, etc. – should be of little concern to the designers who would make better printed parts.

Fused deposition modeling (FDM) is a commercially significant AM technology due to its extensive range of benefits. These includes accessibility, the FDM hardware is usually inexpensive and simple to use; the material flexibility, a wide variety of thermoplastic materials can be used in FDM ranging from inexpensive and readily fluidized grades to sophisticated high technology compositions [11]. These advantages make the FDM process become the ideal solution for designers, hobbyists, and educators who would quickly realize their ideas.

1.3 Use and Limitations of FDM

Additive manufacturing (specifically, FDM) has gained immense popularity owing to its simple operation and inexpensive cost. Also, it gives designers tremendous freedom to create components with freeform and complicated features. However, this flexibility does not give designers total freedom to design any shape they can think of. Like any manufacturing process, AM technologies have their capabilities and their limitations. To determine what are the most challenging problems which constrain designers' freedom for using AM, the principles of design for AM (DfAM) should be considered. These principles are developed on the basis of the limitations of the AM process and guide designers with an intention for minimal violation of constraints.

The principles of DfAM are defined as "a type of design methods or tools whereby functional performance and/or other key product life-cycle considerations such as manufacturability, reliability, and cost can be optimized subjected to the capabilities of AM technologies" [12, 13]. According to the AM limitations, the DfAM methods contain a wide variety of important design considerations that need to be specifically considered during designing. These mainly include the factors which affect the AM process, like part orientation, layer thickness, multi-materials, etc. [14]. Following the methodology provided by Ponche et al. [15, 16], this study focuses on a strategy for designers which treats part orientation and functional and manufacturing path optimization as major steps with which to achieve part quality, functionality and associated mechanical performance. Among these three steps, consideration of part orientation for DfAM implies the most of limitations. These limitations include support structures, staircase effect, and mechanical performance.

Support Structures

As mentioned before, with AM process, each layer is “printed” by means of melted material that accumulates on the consolidated material below and around it. Because of adhesion property of melted material, it can be accumulated between consecutive layers in a manner that is slightly offset. The region of a layer that reaches outward beyond the previous layer is called overhang. Due to the limitation of gravity, material cannot accumulate in the overhang region. In this case, support structures have to be added to the process to ensure that the overhang region is supported. Figure.1.3 shows an example of the overhang region and the related support structures of a flipped L shape in AM. Clearly, we can see from the figure that part orientation has a direct influence on the volume of the support structure required to fabricate the part.

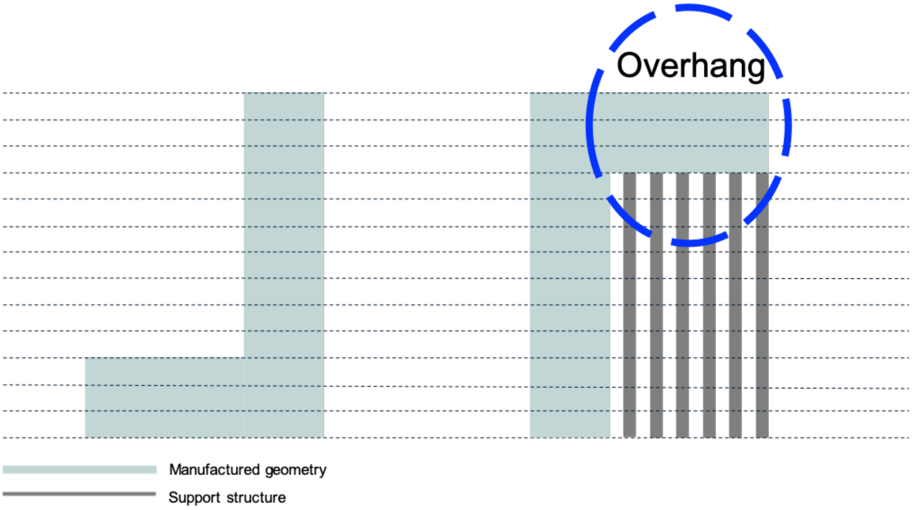


Figure 1.3: Illustration of overhang and the influence of part orientation on the volume of support structures. For the same shape, the appropriate orientation can eliminate the need for support structures (shown in left side).

Support structures have many downsides and impact the whole pipeline of the manufacturing process. These include the following:

- **Increased material cost**: Additional material is used to build support structures, although the material will be later removed and discarded. This material adds to the cost of the fabricated part.
- **Extended building time**: Apparently, more volume has to be printed with support structures. It therefore extends the building time of the manufacturing process.
- **Added post processing work**: Support structures do not belong to the final part. This means that, after printing, the support structure must be removed. This additional task also adds cost to the part due to the labor and time.
- **Damaged surface quality**: Support structures often stick to the surface of the parts. Removing the support structures accordingly results in blemishes or roughness. In the worst cases, the parts are destroyed.

Staircase Effect

The traditional AM process fabricates objects layer-by-layer in a planar manner. As a result, the staircase effect is exhibited because parallel planes are used to approximate a curved surface. Part orientation is important to consider in DfAM, as only horizontal and vertical planar surfaces can be closely matched matched to the design model. All other regions suffer from approximation errors [17]. This issue is illustrated in Figure.1.4, where the wrong orientation for a designed 3D geometry is seen to cause staircase effect in the final fabricated part.

The staircase effect leads to a deviation of the fabricated part from the input model whereby necessary features may be removed. Besides, the surface quality can also be impaired by this phenomenon [18].

Mechanical Performance

The major objective in DfAM is to maximize the functionality [12]. Apart from the design-innovation domain, the functionality of AM parts is also limited by the lagging mechanical performance of AM. Even if the same material is used, the strength of a part made by the FDM process can be a fraction (as low as 10-25%) of the strength from injection modeling [19]. This difference is caused by the fact that in the AM process material is deposited

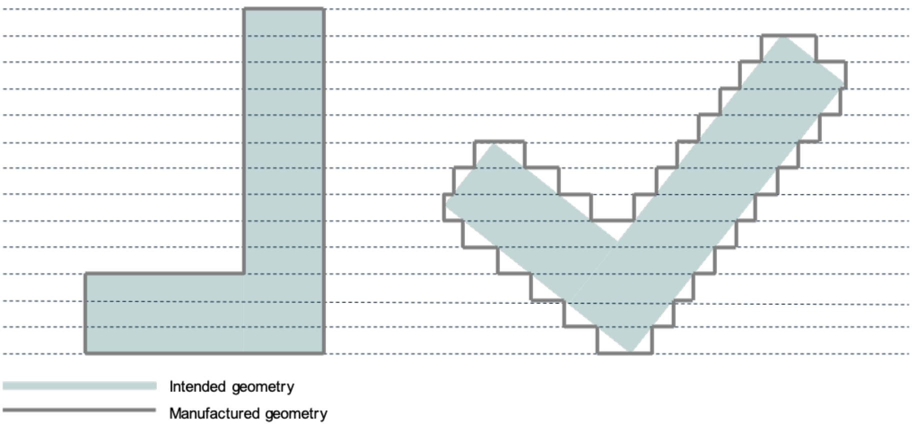


Figure 1.4: Illustration of the staircase effect caused by part orientation. Unlike the inclined shape on the right hand side, the flipped L shape with the right orientation has no staircase effect.

layer by layer along the direction aligning with the part orientation. The resulting layered structure has inherently anisotropic properties, thus causing the tensile strength in-plane (XY-axis) to be much stronger than the build direction (Z-axis), as illustrated in Figure.1.5. The figure clearly shows that part orientation significantly influences the mechanical performance of AM parts.

From the above illustrations, it is clear to see that the major step of DfAM is determining the optimized part orientation that will yield the best trade-offs in terms of minimal staircase effect, support structures and mechanical performance. In addition, other steps have also been devised to reduce these limitations. To improve the mechanical performance and reduce support structures, the functional-optimization step uses topology optimization technology to reforming the structure layout and material density [20]. To diminish the staircase effect, the manufacturing-path-optimization step generates patterned tool-paths for local manufacturing hardware[2].

1.4 Motivation and Research Questions

Although AM is also called 3D printing, it is more appropriately considered “2.5D” printing, as only translation motion is involved during the manufacturing process. In conformity with this manufacturing limit, the

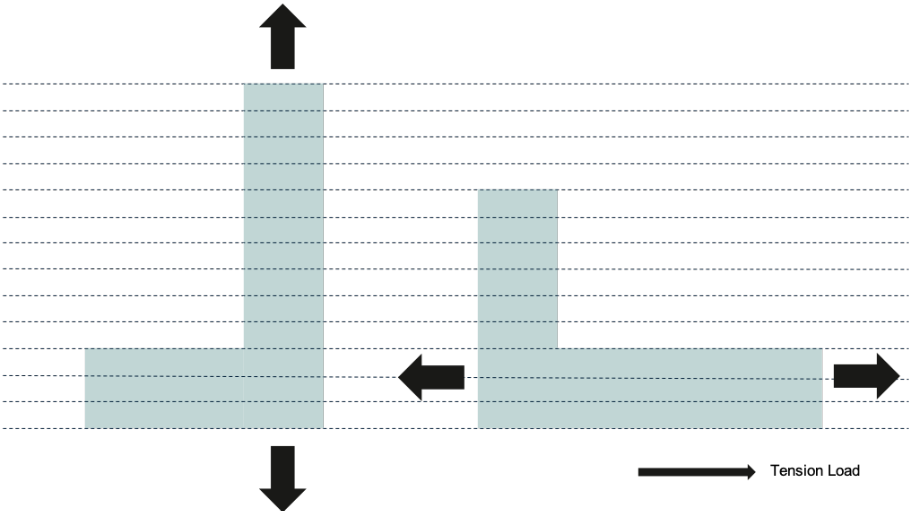


Figure 1.5: Illustration of anisotropic properties in AM. For the same geometry, the image on the left shows what happens when the tension load is normal to the layers: The part is weak. The image on the right shows what happens when the tension load is parallel to the layers: The part is strong.

part orientation should keep being fixed, thereby leading to the problems of staircase effect, support structures, and lagging mechanical performance mentioned in the previous section. Although DfAM methods provide designers with ways to overcome these limitations, most efforts treat the “symptoms” but not the “root cause”. Numerous attempts have been made to optimize the part orientation offline before manufacturing, but few researchers have considered changing the mode of manufacturing itself, allowing a change of orientation to be made during the process of manufacturing [21, 22].

Adding orientation change to the AM process facilitates material accumulation in nearly all directions. This method of manufacturing brings in the possibility of overcoming the above-mentioned problems. Specifically, direct material deposition on the curved surface can reduce the staircase effect caused by planar approximation. Also, overhang regions can be converted to non-overhang region so that the support structures can be eliminated, as shown in Figure.1.6. Last but not least, changing the orientation as per the requirements of loading direction can significantly enhance the mechanical performance. Having made these observations, it is ready to consider the main question of this study:

How to develop an additive manufacturing system with orientation change to push the boundaries of current technology and give product designers more freedom?

In the thesis, an articulated robotic arm with six-degrees-of-freedom (DOF) is adopted (referred to in what follows as a robot) to provide rotational motion in the AM system, thereby allowing part orientation and direction of material deposition to be changed during the manufacturing process. To avoid confusion and make the content more consistent, the term ***robot-assisted AM system*** is used to refer to the AM system with orientation change in what follows. A simple illustration of the robot-assisted setup can be found in Figure.1.6.

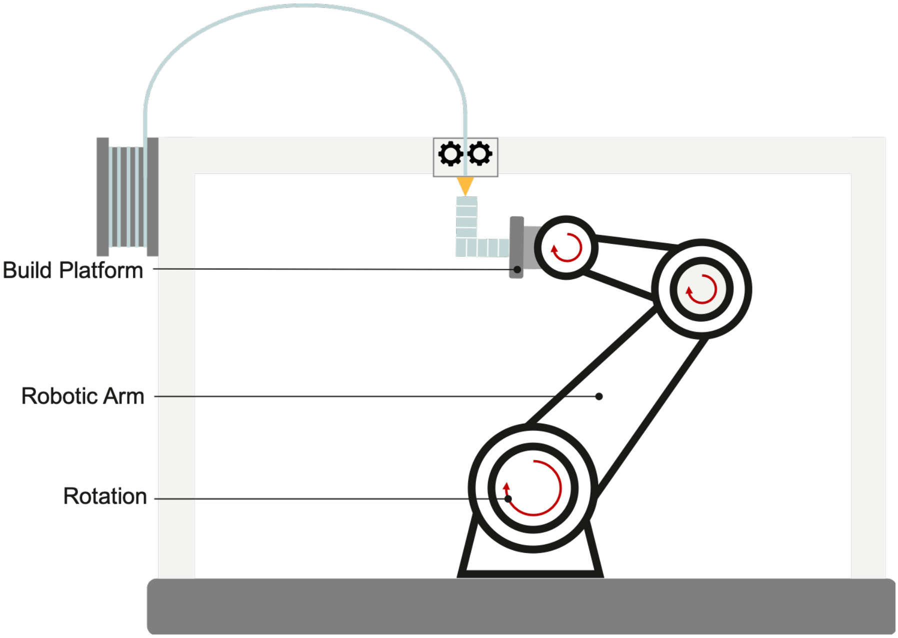


Figure 1.6: A simple illustration of robot-assisted AM. With the rotational motion, the overhang region in the flipped L shape can be converted to non-overhang region so that the support structures can be eliminated.

To reduce the complexity, the research in the thesis follows the workflow of the conventional AM process, but with the aim of fully exploiting the benefits of the rotational motion. A typical AM process workflow includes the following steps:

- **3D Modeling**: The model of the product is first designed in computer-aided design (CAD) modeling software. Most CAD software use Non-uniform rational basis spline (NURBS) patches to present models. The design model needs to be tessellated (usually triangular mesh) so it can be imported to the slicer software and efficiently processed.

- **Slicing**: In the slicer software, the overhang regions are detected, and the software adds support structures to the model and slices them together into a set of planar surfaces with a constant or variable distance between neighbors.
- **Tool-path Generation**: Each slice is converted to a sequence of position points to indicate the movement of the extrusion nozzle. The sequence is called a tool-path.
- **Hardware Realization**: Information regarding tool-paths and extrusion is formatted to the machine language (usually G-Code). The hardware machine receives the G-Code and realizes the commands. After finishing the prescribed tool-path, all planar surfaces are filled with material. The shape of the design model is finally formed.

Figure.1.7 summarizes these steps.

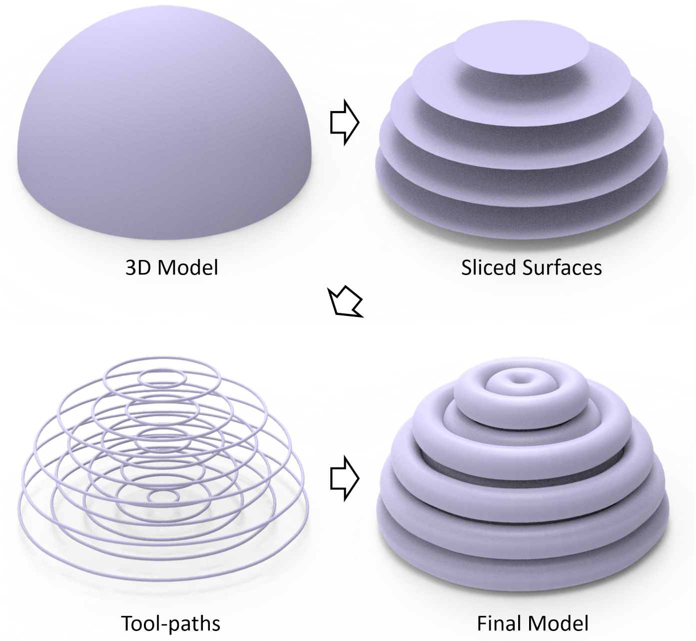


Figure 1.7: The discussed workflow for a conventional AM process.

In an additive manufacturing system with orientation change, the design model can be sliced into freeform surfaces, tool-paths become arbitrary curves in space, and hardware has more complicated configuration space and more physical constraints. So, the main research question can be further elaborated with the following research questions.

RQ1 *How to slice a 3D model in the robot-assisted AM system to overcome the limitations of AM?*

RQ2 *How to generate tool-paths in the 3D domain that can be used in the robot-assisted AM system?*

RQ3 *How to plan the robotic motion in the robot-assisted AM system for better realizing the process?*

RQ1 focuses on the slicing step in the AM workflow. Slicing is the most important question among three, as non-planar slicing leads to changes of orientation which are the key to overcoming the AM limitations. Traditional planar-surface based slicing is a well-defined problem. However, the change from planar to arbitrarily freeform surfaces tremendously increases the complexity of computations. How to define the geometry of a sliced freeform surface while considering manufacturing needs (i.e., fewer support structures) and hardware constraints (avoiding collision between the fabricated part and extrusion nozzle) is challenging and needs to be answered.

RQ2 focuses on the tool-path generation step in the AM workflow. Tool-path generation is relatively simple in the conventional AM workflow because it is performed in the 2D domain. The robot-assisted AM faces the problem to generate evenly spaced, non-overlapping trajectories in the 3D domain while considering the position and orientation continuity for better realization by the robot.

RQ3 focuses on the robot side of the AM workflow. The high-DOF robotic arm is employed to follow 3D tool-paths in the manufacturing process. In other word, planning collision-free motion for a robotic arm to realize the tool-path with feasible dynamical performance is important to ensure the quality of fabrication. However, it is challenging because of the highly nonlinear mapping between robotic configuration space and the waypoint representation in tool-paths. In addition, it needs to consider the kinematic redundancy which results from more degrees of freedom (6-DOF) of the robotic arm than those that are strictly required to execute the tool-paths (5-DOF).

1.5 Research Cycles and Thesis Organization

This thesis is organized around four research cycles (RCs), which are discussed below. Figure.1.8 provides an overview of the research cycles and the corresponding research questions.

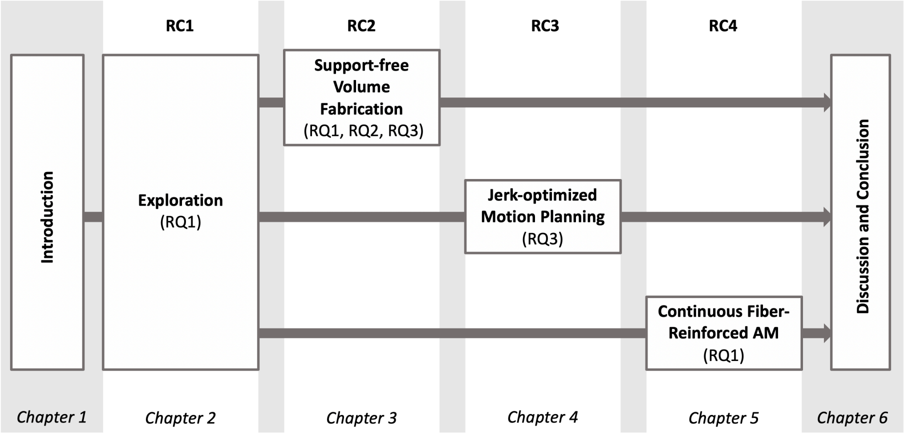


Figure 1.8: Overview of the Research Questions, Research Cycles and Chapters of this thesis.

RC1 The overall objective of the first research cycle is to explore robot-assisted AM. Since robot-assisted AM systems are still immature, and because no commercial product is available with which to realize and validate ideas, prototyping the system is needed to prove the concept of robot-assisted AM. The system was built to develop insights into the research questions and explore opportunities for scientific contributions.

Chapter 2 introduces the first attempt made in this study to use the robotic arm for AM. In this work, models are first subdivided into different regions, then the regions are printed in different orientations. A support-effective volume decomposition algorithm and associated hardware setup are described.

RC2 This research cycle was dedicated to overcoming the limitations found in RC1 and to extending the applicability of robot-assisted AM technology.

Chapter 3 presents a novel approach to decomposing 3D models into freeform surface layers based on the need for support-free and collision-free manufacturing. This slicing method has solved the limitation of the previous work, which can only succeed in simple topology without genus. Moreover, the tool-path generation algorithm for freeform surfaces and the motion planning algorithm for the robot in the AM system are developed to ensure the continuity of manufacturing.

RC3 The third research cycle focuses on the robot side of the system. The goal of RQ3 is explored and formulated.

Specifically, the chapter 4 focuses on the motion planning problem of the robot in the robot-assisted AM system. The relationship between manufacturing quality and the robotic motion is identified in this chapter. As a result, the jerk of motion is optimized to improve the smoothness of robotic motion, thereby improving the overall manufacturing quality. A novel optimization-based motion planning method is proposed to handle the kinematic redundancy of the robotic system and optimize the jerk. Furthermore, a machine learning-based collision detection method is presented to improve the efficiency of computation.

RC4 In the last research cycle, the future shape of AM is explored, which is continuous fiber-reinforced robot-assisted AM.

Specifically, Chapter 5 combines knowledge of robot-assisted AM system with composite material. Exploring the possibility of improving the mechanical performance by realizing the direct deposition of continuous-carbon-fibers in a sandwich structure on freeform surfaces. Physical tests have been conducted to demonstrate the enhancement of the mechanical properties.

Finally, Chapter 6 summaries the contribution of this research project. Unresolved issues and possible future follow-up investigations are described to help interested researchers.

Notice that this thesis is comprised of a collection of conference and journal articles as well as book chapters. The referred publication information is given at the beginning of each chapter.

2

Robot-assisted AM: An Exploration

This chapter presents the first attempt in this research project to add rotational motion to AM. A robotic system – RoboFDM is developed that targets at manufacturing 3D models without support structures, which is considered as one of the major limitations of AM. The hardware of RoboFDM consists of a robotic arm providing 6-DOF motion to the platform of material accumulation and an extruder forming molten filaments of polylactic acid (PLA). The manufacturing of 3D models in this system follows the principle of fused decomposition modeling (FDM). Different from conventional FDM, part orientation of an input model changes during the manufacturing process. A new algorithm is developed to decompose models into support-free parts that can be manufactured one by one in a collision-free sequence. The orientations of all parts are also determined during the computation of model decomposition. Experiments have been successfully taken on the presented RoboFDM system to manufacture general freeform objects in a support-free manner.[1]

2.1 Introduction

In all commercial AM systems, materials are accumulated along a fix direction (i.e., the direction of gravity). The relative motion between a working platform and an extruder is always in the form of translation. The motivation of this work is to provide three additional *degrees-of-freedom* (DOF) so that the

---

[1] This chapter has been published as: Chenming Wu*, **Chengkai Dai***, Guoxin Fang, Yong-Jin Liu, and Charlie C.L. Wang,“RoboFDM: a robotic system for support-free fabrication using FDM”, *IEEE International Conference on Robotics and Automation (ICRA)*, pp.1175-1180, Singapore, May 29 - June 3, 2017 (* Joint First Authors) Note: a few small corrections and/or clarifications have been made to the original published text.

Figure 2.1: A bunny model fabricated by our RoboFDM (left) vs. the model made by conventional FDM (right). Note that, to illustrate our decomposition algorithm, different parts of the bunny model being fabricated along different directions are fabricated by filaments in different colors.

direction of material accumulation can be changed during the fabrication. Our system uses a robotic arm to provide 6-DOF motion on the manufacturing platform. As a result, models can be fabricated in a support-free manner (see Fig.2.1 for an example). To prevent the bad adhesive quality of materials accumulated by a rotational extruder (e.g., the one used in [23]), the extruder is fixed in our system so that the speed of material accumulation can be well controlled with the help of gravity. This means that the function of accumulating materials along different directions is realized by moving the working platform in an inverse way. Figure 2.2 shows the process of fabricating a freeform model by using our system. Note that, although FDM-based AM process is practiced in our implementation, this method can also be applied to other types of AM.

Besides of hardware system, the most challenging issue is how to efficiently and effectively generate a sequence of material accumulation for a given model. Although more DOFs have been gained after introducing a robotic arm into the system, there are still constraints on manufacturability (i.e., the accessibility by collision-free poses), which need to be incorporated during the sequence planning. To better capture the global shape and the topology of an input model, we develop an approach based on shape analysis to generate the sequence of AM process in this work.

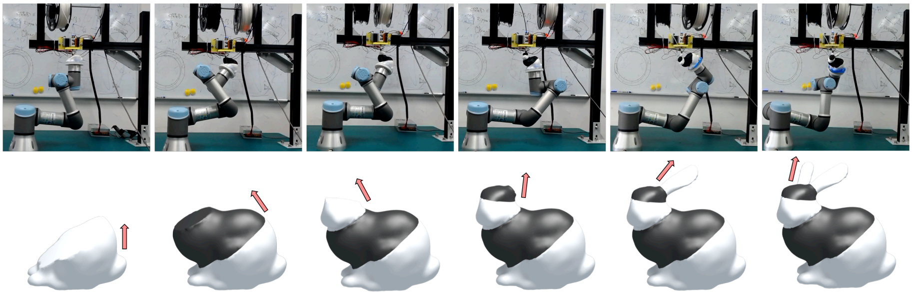


Figure 2.2: The progress of using our RoboFDM system to fabricate a freeform model – bunny. Different parts of the model is fabricated along different directions (see the red arrows in illustration), and filaments in different colors are used for making different parts.

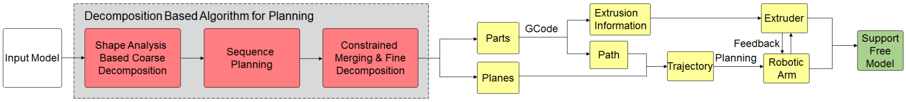


Figure 2.3: Diagram to show the pipeline of our robotic system for FDM-based support-free fabrication. The major contribution of this work is a decomposition based approach for fabrication-sequence planning and the system for realizing support-free FDM on a robot-assisted system.

2.2 Related Work

To overcome the limitations of layer-based AM as mentioned in previous chapter, research has been taken in the aspects of 1) adding more DOF into motion and 2) optimizing shape (or direction) of fabrication [1].

More and more researchers are exploring the new DOF of AM in motion so that the process of manufacturing can be improved. Keating and Oxman [22] present a manufacturing platform using 6-DOF provided by a robotic arm to fabricate models in both additive and subtractive manners, where the concept of freeform fabrication without supporting structures is proposed. However, only very simple shapes are fabricated in their approach and no detail of sequence and path planning is provided. Pan et al. [24] proposed a 5-axis motion system similar to 5-axis CNC machining to accumulate materials onto an existing model, where the algorithm can only deal with small components with a relative simpler planning task. A 6-DOF parallel kinematic Stewart platform is presented in the work of Song et al. [25] for the purpose of multi-directional AM. Again, only small components with simple shapes are processed. There is no algorithm for fabricating general freeform models. In the recent work called RevoMaker [21], freeform models are fabricated by AM on top of an existing cubic component, where simple shapes can be fabricated. The cuboidal platform used in their system cannot print general freeform models like the bunny shown in Fig.2.1 without supporting structures.

On the other aspect, layer-based manufacturing is still a widely used method in commercial AM systems, which however needs to use additional supporting structures to avoid collapse on overhangs. A few approaches have been developed to reduce the usage of supporting structure by either model deformation or model decomposition. Hu et al. [26, 27] proposed a deformation based method that modifies the shape of input by using local minimal rotations to eliminate 'facing down' regions. Differently, in our system rotations are introduced during the fabrication process so that the shape of a model does not need to be changed. Segmentation methods are introduced in [28] and [29] to decompose an input model into pieces in the shape of height-fields. The physical model is fabricated by printing the height-field parts one by one and assembling together by glue. These segmentation methods cannot be directly applied here as collision in the process of fabrication are not considered, which will be solved in our approach.

2.3 System

We present a robotic solution for support-free fabrication using FDM-based material accumulation. The overview of our system is introduced below.

2.3.1 Hardware Setup

The hardware setup of our system is mainly composed by a UR3 robotic arm, a FDM extruder fixed on a frame and all other control components. To demonstrate the functionality of fabricating multi-materials, white and black PLA filaments are used. Different regions on a model can be fabricated by using filaments in different colors. The extruder is fixed in our system, and the relative material accumulation direction and position is realized by inverse pose of the manufacturing platform attached on a 6-axis UR3 robotic arm. The UR3 Robotic arm is used to provide 6-DOF motion during the process of material accumulation. Considering the accuracy of positioning that can be achieved on a UR robotic [30] and the speed of fabrication, nozzles with 1mm or 2mm holes are employed in our system. Generally, 2mm nozzles are used for quick fabrication and 1mm nozzles can fabricate models with better surface quality and more geometric details.

The relative position of nozzle to the base frame of the robotic arm is also very important as it will be a problem if there are many inaccessible positions and orientations in the working envelope defined around the nozzle. The position to place nozzles should be optimized according to the reachability analysis of the robotic arm. First of all, we randomly sample the configurations space of UR3's joints and determine the position and orientations of each configuration point in the Euclidean space. Considering the fixed orientation of the nozzle, all samples with facing-down frames are excluded. The remaining frames are considered as reachable samples. The working envelope around a nozzel is defined as a bounding box $\mathcal{B}$ centered at the nozzle. The optimal position of a nozzle can then be determined by exhaustive search to find a place such that the number of frames falling in $\mathcal{B}$ is maximized.

2.3.2 Software Method

The basic idea of our work is to rotate the working platform to a 'good' orientation for each segmented part so that layer-based material accumulation for this part can be completed without adding support. Specifically, a 3D model is clipped into smaller parts by planes (called *base plane* in the rest of this chapter). The input model can then be fabricated following the sequence of connected parts, and the surface of each segmented part is self-supported when being fabricated along the normal direction of its base plane. To generate the segmentation, we propose a coarse-to-fine decomposition algorithm which first segments a model into multiple parts according to the skeleton based shape analysis (i.e., each branch a part). After that, the sequence of 'growing' is determined by incorporating the collision-free constraint of our hardware system. Finally, a fine-level partition is applied to each part to refine the sequence of manufacturing. With the help of this decomposition algorithm, the path planning of our robotic FDM printer can be realized. For each part, layers are generated by slicing planes parallel to the part's base plane, and G-code of tool paths can be generated in the same way as conventional FDM.

The low-level software components for controlling the operation of our system are based on the ROS system. Specifically, an Arduino-based MKS Gen Board is used to control the temperature of the hot-end and also the step motor for feeding materials in each extruder. The Arduino board is set as an ROS node for communicating with UR3. In our implementation, an ROS-based driver for UR robot has been extended to add a trajectory

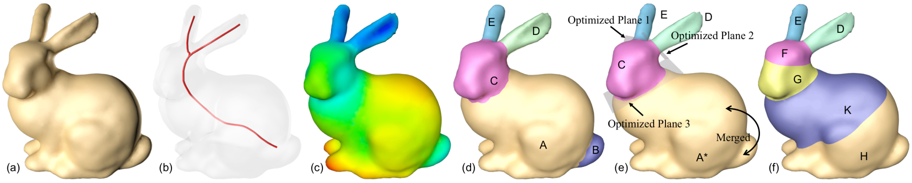


Figure 2.4: Progressive results of our algorithm on the bunny model: (a) input, (b) extracted skeleton, (c) distribution of the shape diameter metric, (d) the result of coarse segmentation, (e) the result after plane perturbation and merging, and (f) the final result after fine decomposition.

feedback function so that operations on the robotic arm and the extruder can be synchronized.

2.4 Algorithms

In AM, a model $\mathcal{M}$ is fabricated layer by layer along a fixed direction $\mathbf{d}$. Borrowing the definitions of maximal self-support angle α_{max} and the self-supported region presented in [26], a face on $\mathcal{M}$ with its normal $\mathbf{n}$ satisfying

$$\mathbf{n} \cdot \mathbf{d} + \sin(\alpha_{max}) \geq 0 \tag{2.1}$$

is called *safe* face; otherwise, it is a *risky* face. A segmentation based method is developed to eliminate risky faces.

Problem Definition: Given a model $\mathcal{M}$ to be fabricated by RoboFDM, we determine a decomposition of $\mathcal{M}$ as $\{\mathcal{M}_i\}$ $(i = 1, \ldots, n)$ with

$$\mathcal{M}_1 \cup \mathcal{M}_2 \cup \cdots \cup \mathcal{M}_n = \mathcal{M}$$

so that an optimal direction $\mathbf{d}_i$ can be computed for each segment $\mathcal{M}_i$ to make all regions on its boundary surface $\partial\mathcal{M}_i$ safe and all components can be fabricated in a *collision-free* manner one by one.

A decomposition approach with three phases is developed to solve this problem, and constraints to avoid collision are incorporated into our decomposition algorithm.

2.4.1 Phase I: Shape-analysis-based Coarse Decomposition

In the first phase of our algorithm, a skeleton based analysis is taken to generate the first decomposition of $\mathcal{M}$ so that can be decomposed into branches with each part having a simple topology (see Fig.2.4(a)-(d)). Specifically, the following steps are taken in our algorithm:

1. A mean-curvature flow based algorithm [31] is firstly applied to extract the 1D line skeleton $\mathcal{S}_{\mathcal{M}}$ of $\mathcal{M}$. The number of branches, k, is determined (e.g., $k = 3$ for the bunny example shown in see Fig.2.4(b)).
2. For every point $\mathbf{p}$ on the surface $\partial\mathcal{M}$, a shape diameter metric $D(\mathbf{p})$ is evaluated by its distance to the closest point on $\mathcal{S}_{\mathcal{M}}$ (see Fig.2.4(c)).
3. An expectation maximization algorithm is conducted to fit k Gaussians to the histogram of $D(\cdot)$ on $\partial\mathcal{M}$ to first have a soft-clustering of faces to k clusters based on the values of their shape diameter metrics. This is followed by a hard partition of faces using k-ways graph-cut to consider local geometry on $\partial\mathcal{M}$ (Details can be found in [32]).

Note that, different from the original approach presented in [32], a 1D line skeleton computed from mean-curvature flow is used here to enhance the robustness of partition. Applying one iteration of Laplacian smoothing to $\mathcal{M}$ before the above three steps can further enhance the robustness of the algorithm.

Boundaries of surface regions obtained from above method are not planar, which are hard to be used in the layer-based AM. We generate the decomposition of $\mathcal{M}$ by fitting and adjusting the separating planes between different parts. Specifically, we first sample the boundary curve between two segmented patches into points, and their inward/outward offset points can be generated along the surface normals with a very small offset value. Principal component analysis is then applied on all these points to obtain a cutting plane approximating the boundary between two patches. These planes now form a coarse decomposition of the input model, $\mathcal{M}$.

2.4.2 Phase II: Sequence Planning

To build the sequence of fabrication, the result of a coarse decomposition can be first converted into an undirected graph $\mathcal{G}$ by using the neighboring information between parts. Specifically, each part $\mathcal{M}_i$ is considered as a node in $\mathcal{G}$. When $\mathcal{M}_i$ and $\mathcal{M}_j$ are connected, an undirected edge is constructed in $\mathcal{G}$ to represent this neighborship. For example the bunny model shown in

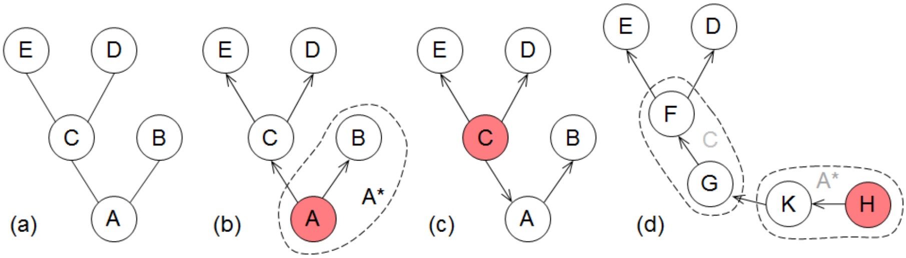


Figure 2.5: The graph representation of the bunny model for sequence planning: (a) an undirected graph as the result of coarse decomposition, (b) a directed graph if the sequence start from node 'A', (c) another possible directed graph with node 'C' as the root, and (d) the resultant graph after applying the merging and fine decomposition steps in Phase III of our algorithm. 'A' and 'B' are merged into a new node 'A*', which is further split into 'K' and 'H' in the fine decomposition step. The nodes 'F' and 'G' in (d) are split from 'C'. Corresponding region of each node can be found in Fig.2.4.

Fig.2.4, its undirected graph can be defined as shown in Fig.2.5(a). Starting from a selected node (e.g., node 'A' in Fig.2.5(b)), we can generate a sequence of nodes on the graph by a breadth-first-traversal. The order of visit gives the directions of edges – i.e., $\mathcal{G}$ is converted into a directed graph $\bar{\mathcal{G}}$ with the starting node named as *root*. Note that, this conversion depends on the starting node of search. When starting from a different node, a different graph can be generated (e.g., the directed graph with node 'C' as a root is shown in Fig.2.5(c)). In our current implementation, the root is interactively selected by users.

After getting the sequence of fabrication based on the coarse decomposition, the orientation of fabrication can be determined according to the planes separating different parts of $\mathcal{M}$. When two parts $\mathcal{M}_i$ and $\mathcal{M}_j$ are separated by a plane $P_{i,j}$ and have the fabrication sequence of $\mathcal{M}_i$ followed by $\mathcal{M}_j$, the process to fabricate $\mathcal{M}_j$ will start from accumulating materials on the plane $P_{i,j}$. Defining the normal $\mathbf{n}_{i,j}$ of $P_{i,j}$ in an orientation pointing from $\mathcal{M}_i$ to $\mathcal{M}_j$ (with $\mathbf{n}_{j,i}$ having an inverse orientation), the material accumulation direction of $\mathcal{M}_j$ is heuristically given as $\mathbf{n}_{i,j}$. Taking this heuristic to determine the material accumulation direction of each component, we need to further adjust the separating planes in the last phase of our algorithm. The oriented plane $P_{i,j}$ is named as the *base plane* for the fabrication of $\mathcal{M}_j$, and the cross-section $P_{i,j} \cap \mathcal{M}$ is called the *base cross-section*. Besides, the material accumulation direction of root $\mathcal{M}_r$ is determined by finding an optimal one resulting in a minimal facing down area while keeping all

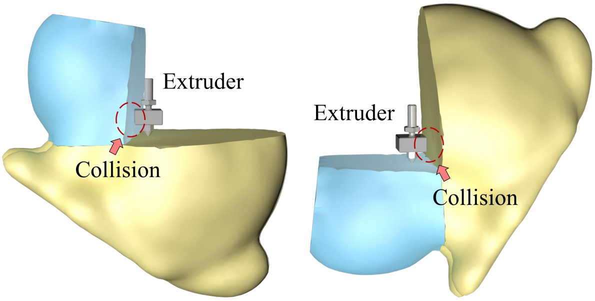


Figure 2.6: Collision between the nozzle and the fabricated model occurs when the base cross-sections intersect with each other – changing the sequence of fabrication cannot resolve the problem.

separating planes at the boundary of $\mathcal{M}_r$ facing up.

2.4.3 Phase III: Constrained Fine Tuning

With the material accumulation direction determined from the base cross-section of a part $\mathcal{M}_i$, the risky region of $\mathcal{M}_i$ can be defined as the surface regions on $\partial\mathcal{M}_i$ (excluding the base plane) that do not satisfy Eq.(2.1). The area of risky region, $R(\cdot)$, is a metric used in our constrained fine tuning. A completely support-free component $\mathcal{M}_i$ should have $R(\mathcal{M}_i) = 0$. The constraints to prevent collision are:

- *Constraint I* – The cross-sections formed by the separating planes (i.e., the base cross-sections) should not intersect with each other (as illustrated in Fig.2.6).
- *Constraint II* – All base planes need to face up.

In this phase of our algorithm, the decomposition is further refined by applying 1) plane perturbation, 2) region merging and 3) fine decomposition. Each part of the decomposed model will be ensured to be support-free, and the heuristic constraints for avoiding collision will be incorporated.

Plane Perturbation The separating planes determined by PCA in the phase of coarse decomposition will be further optimized to enhance the manufacturability of each part. For a plane $P_{i,j}$ separating $\mathcal{M}_i$ and $\mathcal{M}_j$, small random perturbation is added to its position (with offset less than 2.5% of the bounding-box's diagonal length) and orientation (within 5° angular variation). The areas of cross-sections formed by the perturbed planes are computed, and those lead to more than 20% area increase in cross-section will be excluded. The total area of risky regions on $\mathcal{M}_i$ and $\mathcal{M}_j$ is evaluated as $R(\mathcal{M}_i) + R(\mathcal{M}_j)$. A separating plane leads to the smallest value is considered as optimum to be used.

Region Merging As the coarse decomposition based on shape analysis did not explicitly consider the area of risky region (i.e., manufacturability), it is possible that a component merged from two neighboring parts has smaller risky region. The parts 'A' and 'B' of the bunny example shown in Fig.2.4 are a good example – merged into a new part 'A*' (see also the graph representation in Fig.2.5). Base plane that is visited earlier in the determined fabrication sequence will be used as the new base plane. For example on the bunny model, the base plane of 'A' will be used as the base plane of 'A*'.

Fine Decomposition Nodes on the directed graph representing the decomposition will be further refined to ensure the manufacturability. A greedy strategy is applied here. The component with largest risky area, $\mathcal{M}_g$, is selected to be clipped into two components, $\mathcal{M}_g^s$ and $\mathcal{M}_g^t$, by an optimal plane that can minimize the total area of risky region and satisfy the constraints for preventing collision. After applying this refinement, the node $\mathcal{M}_g$ on the direct graph and in the sequence of fabrication will be replaced by $\mathcal{M}_g^s$ and $\mathcal{M}_g^t$. The refinement on the most risky region is repeatedly applied until all components become safe (see the result in Fig.2.4 and Fig.2.5(d) as an example). In our implementation, we select k regions from the top of queue ($k = 6$) instead of only the most risky one to avoid being stuck at the local minimum during the refinement.

Optimal Cutting Plane All possible planes $\pi_{k,l} = (\mathbf{n}_k, d_l)$ with planar equation $\mathbf{n}_k \cdot \mathbf{x} - d_l = 0$ that satisfy constraints I and II are considered. The upper half of the Gaussian sphere is uniformly sampled into 100 points for the possible values of $\mathbf{n}_k$, and the plane offsets $\{d_l\}$ are also sampled uniformly at intervals of 5mm in our implementation. This follows the same sampling strategy of clipping planes in Chopper [33]. Among all these planes, the optimal plane for clipping is defined as

$$\arg\max_{\pi_{k,l}} (R(\mathcal{M}_g) - (R(\mathcal{M}_g^s) + R(\mathcal{M}_g^t))) \tag{2.2}$$

with $\mathcal{M}_g^s$ and $\mathcal{M}_g^t$ being the two sub-parts clipped from $\mathcal{M}_g$ by $\pi_{k,l}$ (i.e., $\mathcal{M}_g^s \cup \mathcal{M}_g^t = \mathcal{M}_g$ and $\mathcal{M}_g^s \cap \mathcal{M}_g^t = \pi_{k,l}$).

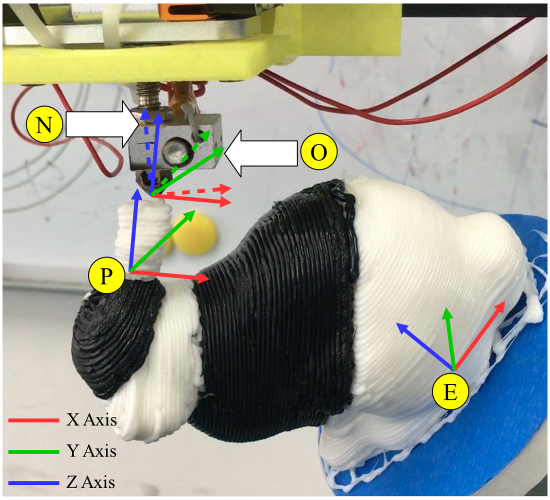


Figure 2.7: Frames used in the motion control of our RoboFDM system.

2.5 Implementation Details and Results

The decomposition based planning algorithm presented above has been implemented in a C++ program to generate mesh models for each component. The slicing software for conventional FDM is used to generate planar slices and tool paths according to the material accumulation directions determined in our algorithm. G-code for AM can be generated from the planned paths and sent to the motion-control module of our system.

The motion of UR3 is realized by first determining the pose of its end-effector according to a given position and orientation for material accumulation in the Euclidean space. As illustrated in Fig.2.7, there are five different frames in our system including:

- *B*: Frame of the base link on the robotic arm, which is also set as the origin of world-coordinate system;
- *E*: Frame of the end-effector with origin located at the center of the first base plane to fabricate;
- *P*: Frame for the current part to be fabricated with origin located at the center of the current base plane for fabrication;
- *O*: Frame at the current position and orientation to accumulate material;
- *N*: Frame of the FDM extruder with origin at the end of nozzle.

Figure 2.8: More examples that are fabricated by our RoboFDM system – (left) Armadillo and (right) Hanging-Ball without adding any supporting structure.

Table 2.1: Statistic of Fabrication

Model	Bunny	Armadillo	Hanging-Ball
Number of Facets	10,000	17,142	5,984
Decomposition Time	171 sec.	269 sec.	156 sec.
Fabrication Time	152 min.	263 min.	85 min.

The relationship among these frames can be defined as

$$\mathbf{T}_E^B \mathbf{T}_P^E \mathbf{T}_O^P = \mathbf{T}_N^B, \tag{2.3}$$

where $\mathbf{T}_N^B$ is a constant matrix as the extruder is fixed in our system, $\mathbf{T}_O^P$ is known according to the tool-path represented by G-code, and $\mathbf{T}_P^E$ can be updated during the fabrication process. With these known transformations, $\mathbf{T}_E^B$ can be obtained by solving Eq.(2.3). After determining $\mathbf{T}_E^B$, inverse kinematics is applied to compute a corresponding configuration in the joint space of UR3. The motion between poses is generated by using the *RRT-Connect* based planning [34], with which collision can be avoided.

We have tested our system by fabricating a variety of models in a support-free manner. Figure 2.3 has shown the whole process of fabricating a support-free bunny model, where supporting structures must be added below the head and at the ears by using conventional AM process(see the right of Fig.2.1). Figure 2.8 shows our results of decomposition and fabrication on two other models. Some statistics of computation and fabrication are given in Table 2.1.

At the same time, we also tested the mechanical properties of a model fabricated by RoboFDM and compared it with the same model fabricated by a conventional AM system. As shown in Fig.2.9, the stiffness of the model fabricated by RoboFDM is weaker, which is mainly caused by the weak adhesion of materials at the interface between two regions. On the other aspect, the delamination of layers happens at nearly the same level of loading (i.e.,

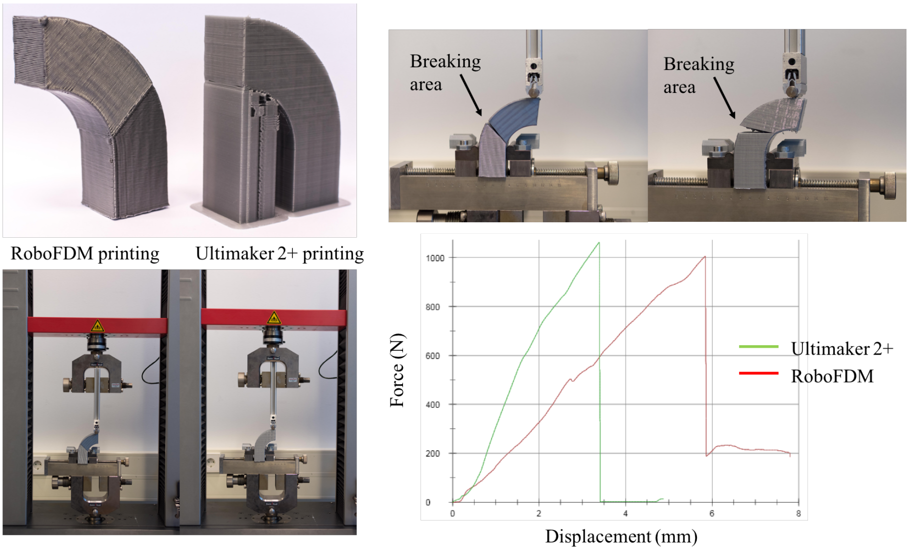


Figure 2.9: Mechanical tests taken on models fabricated by RoboFDM and conventional AM.

around 1000N).

2.6 Conclusion and Discussion

We propose a novel robotic system to fabricate models in support-free AM. The 6-DOF motion is provided to the working platform so that materials can be accumulated along different directions in the working space with a fixed extruder. The core part of our system is a model decomposition algorithm, which segments an input model into parts that can be fabricated without adding supporting structures. The sequence of fabrication is determined together with the decomposition. Our experimental tests give promising results, and the models with large overhangs can be fabricated in a support-free way.

There are still some limitations in our current implementation of RoboFDM, which fall into the following aspects.

1) *Speed*: We set the motion of the robotic arm at a very slow speed to ensure the accuracy of positioning; therefore, the fabrication time in own system is longer than the conventional FDM printer for the same model.

2) *Size*: As the nozzle is fixed in our current setup, the volume of a model to be

fabricated is quite limited by the frames and the table. This can be improved when we add two more DOF to the frame and the nozzle in motion.

3) *Cost of System*: The hardware cost of using a robot arm to fabricate models is much higher than using a conventional 3D printer. On the other aspect, more flexibility of fabrication is given by the RoboFDM system.

4) *More complex shape*: Our decomposition algorithm may fail on models with a high genus number, on which supporting structures still need to be added for fabrication.

5) *Better strategy of collision-free*: The constraint to avoid facing down base planes employed in this work is too restrictive although it works well for preventing collision. As a result, the support free fabrication of some model (e.g., a tree with pointing down branches) may fail.

In summary, this first attempt gives promising results in experiments. As mentioned above, there are still a few challenges need to be tackled. An improved method will be described in the next chapter.

3

Robot-assisted AM: From Planar into 3D

As stated in the conclusion section of Chapter 2, the previous work may fail on the models with complicated geometry. To tackle this challenge, a new method is proposed to slice 3D models with curved layers instead of planar layers. With this new method, the need for supporting structures can be tremendously reduced – if not completely abandoned – on all models. The strategy to tackle the challenge of complicated geometry is to perform two successive decompositions, first volume-to-surfaces and then surfaces-to-curves. The motion planning algorithm has also been developed for the robotic arm to enable continuous motion. In the physical experiments, the work presented in this chapter successfully fabricated the 3D models with large overhangs and high-genus topology. [2]

3.1 Introduction

The previous work presented in Chapter 2 is a modest step in the direction of robot-assisted AM. 3D model is decomposed into several parts and fabricating each part with different orientations. The decomposition and orientation are represented by a sequence of planar clipping planes, and the orientation only changes between the printing of each decomposed part. Although this method can significantly reduce the usage of support structures for simple geometry models, there exists no solution for complicated geometry to generate a support-free decomposition. To overcome this limitation, the slicing

---

[2] This chapter has been published as: **Chengkai Dai**, Charlie C.L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, Yong-Jin Liu, "Support-Free Volume Printing by Multi-Axis Motion", *ACM Transactions on Graphics (SIGGRAPH 2018)*, Vol. 37, No. 4, Article No.1, 2018. Note: a few small corrections and/or clarifications have been made to the original published text.

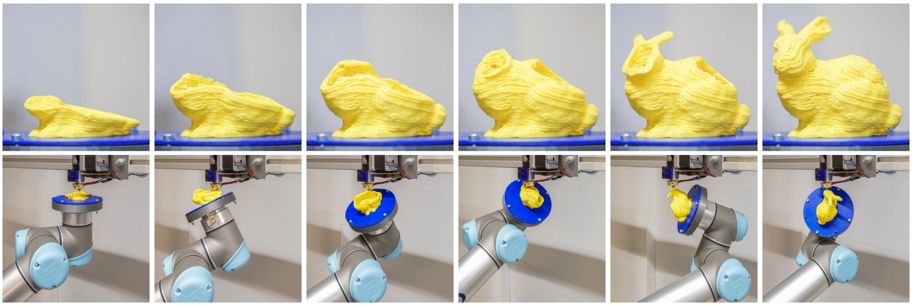
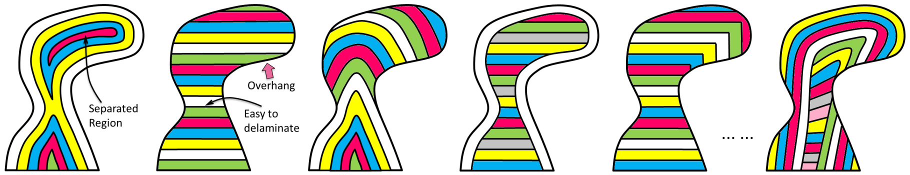


Figure 3.1: The method proposed in this chapter enables fabricating solid models support-freely. By exploiting all 6 degrees of freedom (translations, rotations) and depositing material along *curved* layers, the support structures become unnecessary in most cases. This increases further the flexibility offered by AM such as freeing designers from support constraints on complex parts.

is changed from planar into 3D domain in this work, allowing continuous orientation change during fabrication and exploiting all the benefits of high DOF motion.

Compared to conventional AM, this change from planar to arbitrarily curved layers tremendously increases the complexity of computations. While constraining layers to be planar leads to a well defined slicing problem, the additional freedom introduced by curved layers makes it challenging to even define what the geometry of the layers should be. In addition, a feasible solution has to take into account geometric constraints as well as hardware constraints. A detailed illustration is shown in Figure.3.2. Among all listed decompositions, the 'micro'-structure of Figure.3.2(a) is conformal to the boundary surface of a given solid. However, separated regions in Figure.3.2(a) are impossible to be fabricated even when a robot-assisted AM device is used. Alternatively, the process planned in Figure.3.2(c) can be physically realized. When manufacturing a model by using the layers in Figure.3.2(b), supporting structures need to be added below overhangs. A better appearance with less staircase artifacts might be obtained from the decomposition as shown in Figure.3.2(d). With the help of multi-axis motion, supporting structures can be avoided by using the curved layers shown in Figure.3.2(e). The purpose of our work presented in this chapter is to compute such a feasible solution to fabricate a given solid model by minimal supporting structures.

In the following content of this chapter, we present the new methodology to tackle the challenge of slicing, tool-path generation and hardware realization. Our technique is based on the observation that the dimensionality of the



(a) Conformal structure (b) Planar layers as conventional (c) Semi-conformal Structure (d) Structure with less staircase artifacts (e) Support-free structure

Figure 3.2: There are many choices for decomposing a volume into curved layers with nearly uniform thickness for tool-path generation – here different layers are shown in different colors.

problem can be successively reduced by first decomposing the volume into sequences of curved surface layers (slicing), and then decomposing each surface into curved tool-paths. Our algorithm searches for an accumulation sequence, which is collision-free, ensures always supported material deposition, and can print all regions as much as possible. Curved surface layers are covered with tool-paths taking into account hardware constraints.

Contributions:

- A novel approach for robot-assisted AM, that decouples the problem into first extracting curved surface layers, and second covering each surface with curved tool-paths, successively reducing the dimensions of the problem.
- An algorithm based on the computation of a scalar field representing the accumulation sequence of material within the shape during the AM process. The field is carefully constructed such that layers are convex and collision-free, supported by previous layers, and – as much as possible – do not prevent *future* layers to be accessed.
- The covering of curved layers with smooth tool-paths, optimizing both positions and orientation according to the constraints of the 6-DOF robot realizing the motions.

We demonstrate our approach on a variety of models in both computational and physical experiments, fabricating actual objects using the tool-paths generated by our approach to drive a 6-DOF filament based AM platform.

3.2 Related Work

Since its invention in the late 1980s, the core principle of AM remained largely based on the accumulation of planar layers along a single build direction [35]. Active research in the area focuses on the use of multi-materials, faster speed and increased deposition flexibility [1]. In particular, a recent trend is to exploit additional motion DOFs, moving away from the current limitations of planar material accumulation.

3.2.1 3-DOF AM System

The first attempt of using non-planar layers in AM was made a decade ago in an approach called the *Curved Layer Fused Deposition Modeling* (CLFDM) [36]. It departs from standard FDM fabrication by dynamically changing z-values within individual layers. Recently, such motions have been realized on a Delta style FDM printer [37]. A shell model is fabricated by depositing a double-curved layer on top of a sandwich structure fabricated with planar layers. As a result, the surface of the 3D printed model does not exhibit the staircase effect. However, this approach is limited to models with relative simple shapes – i.e., height-fields facing up along the z-axis. In addition, regions with a steep slope lead to local collision between deposited materials and the extrusion head. A most recent effort was paid to generate 3-axis motion tool-paths inside a given volume [38], which also suffered from gouging.

3.2.2 High-DOF AM system

Keating and Oxman [22] introduced a FDM based proof-of-concept AM, showing how exploiting all 6 DOFs of a robotic arm can improve the AM process. The demonstration is however limited to simple shapes (e.g., cubes, torus and cylindrical surfaces) and there are no details regarding tool-path generation. Interestingly, the extrusion nozzle is fixed while the robot moves the part below. We use a similar setup (see supplemental material). This increases filament adhesion with the help of gravity, in contrast to moving an extrusion head around a fixed part (e.g., [39]).

Pan et al. [24] developed a five-axis motion system similar to CNC machining, accumulating materials onto an existing model. The tool-path planning algorithm only handles specific cases and relatively simple components.

Recently, researchers have focused on fabricating wire mesh models using high DOF. Such models are fabricated edge by edge, using freeform motions. Wu et al. [40] compute collision-free tool paths for this purpose. A naïve ordering can lead to a configuration where some edges cannot be approached anymore. To tackle this challenge, a global sequence planning is formulated on a directed graph. Huang et al. [41] further considers stability constraints jointly with the collision-free constraints. Both these approaches detect collisions in the optimization loop, which is time-consuming. As a result, only wire meshes with small number of primitives can be considered (e.g., less than 1k struts in [41, 40]). This drawback prevents applying these algorithms to large scale problems (e.g., the Bunny model in Fig.3.1 has 97.5k voxels with 0.8mm width – relevant to the nozzle's diameter).

High-DOF robotic devices have been extensively used in composite fabrication (e.g., aeronautical industry [42]). However, specifying the tool-paths for placing composite tapes on curved surface often requires an intensive manual work. Our ambition is to automate the tool-path generation for high-DOF AM on general models, which is a critical step for direct digital manufacturing. None of the existing approaches investigates a method to produce curvilinear tool-paths within the volume of a part.

3.2.3 Volume Decomposition for AM

In another thread of research, volume decomposition has been used to enable the AM process in different scenarios. Luo et al. [33] decompose large models into smaller pieces so that they can be fabricated on 3D printers with limited working envelopes. Other methods decompose a given model into height-fields [28] or pyramid-based shapes [29] so that they can be fabricated by molding or support-free AM. To fabricate large models, Song et al. [43] decompose a volume into a set of large-core-supporting height-field pieces that are 3D printed. These approaches require a manual assembly step and the final parts present fragilities along assembly surfaces.

Rotational motions have also been used to avoid manual assembly. In [21], material accumulation is applied around a cubic component, fabricating a 3D model on top of an existing object.

Our previous work [44] in Chapter 2 proposes an algorithm to segment a model into support-free parts, each fabricated by a robotic arm using planar layers.

All these methods still rely on planar layers, which constrains both the decomposition and the complexity of the parts that can be handled. In this chapter, we investigate a more general curved layer decomposition method for high-DOF AM.

3.2.4 Accessibility for Machining

Determining accessibility remains a challenging aspect of multi-axis tool-path planning, despite its extensive study in the context of CNC milling. Algorithms include the visibility map [45] to analyze accessibility, as well as approaches that detect and avoid collisions between tools and workpieces [46]. Recent work focus on computing gouging-free tool-paths (i.e., no over-cut caused by local interference between tool and workpiece) while also optimizing the dynamic behavior of machines [47, 48]. The surface accessibility of a given model has also been widely studied in other areas such as for molding [49], and for inspection and computer-assisted surgery [50].

In general, the computation of accessibility is very time-consuming. Our problem is even more challenging – we seek to decompose a volume into a *sequence of accessible* surfaces with nearly uniform thickness.

Our attempts at using existing state-of-the-art collision detection techniques (i.e., the *Flexible Collision Library* [51]) were discouraging.

For example, the candelabra model in Fig.3.3 has 186,735 voxels using a voxel dimensions of 0.8mm, based on the nozzle diameter. Given a sequence that adds voxels one by one, the collision detection step alone – that incrementally adds voxels and checks for collisions – can take up to *96 hours* in total. However, to find a support-free tool-path a large number of such possible sequences have to be checked. A brute force approach could not be computed in any feasible amount of time. Instead, we propose a new method to *maintain* an accessible working surface while progressively constructing the sequence of material accumulation.

In summary, advanced AM hardware capable of multi-axis motions cannot be fully utilized at present, for lack of effective tool-path planning algorithms. Although the techniques developed for CNC milling are relevant, going from surface machining to volume filling for AM tremendously increases the complexity and the difficulty of the related geometric problems.

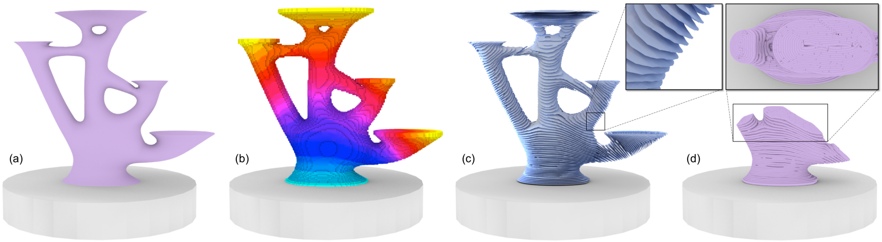


Figure 3.3: Illustration of dimensionality reduction for the process planning of robot-assisted AM. From left to right: (a) input solid $\mathcal{H}$ for a topology optimized candelabra, (b) accumulation field, (c) curved layers $\{\mathcal{S}_i\}$ extracted from $\mathcal{H}$ and (d) curved tool-paths $\{\mathcal{P}_j\}$ covering each curved layer.

3.3 Method Overview

To tackle the challenge of slicing and tool-path generation for support-free robot-assisted AM, we propose a novel approach based on a *dimension reduction* strategy. As illustrated in Fig.3.3, a given model will be first decomposed into a valid sequence of curved layers that are manufacturable (slicing), which are then further decomposed into curved tool-paths.

There are many possible choices for the decomposition in curved layers, as well as for covering each layer with curves. We aim at effectively finding feasible solutions for these two problems, while taking into account manufacturability constraints.

3.3.1 Decomposition in curved layers

We formulate the problem as follows. Given a solid model $\mathcal{H}$, we seek to decompose it into a sequence of (curved) surface layers $\{\mathcal{S}_i\}_{i=1,\ldots,n}$, such as to represent the material accumulation in AM. This requires satisfying the following conditions:

1. The solid $\mathcal{H}$ is well approximated by the curved layers as $\mathcal{H} \approx \cup_{i=1,\ldots,n}\Pi(\mathcal{S}_i)$ with $\Pi(\mathcal{S}_i)$ denoting the convolution solid of $\mathcal{S}_i$ by a sphere with diameter r (layer thickness), and there is no overlap between layers – $\Pi(\mathcal{S}_i) \cap \Pi(\mathcal{S}_j) = \emptyset$ $(\forall j \neq i)$;
2. All surface patches $\{\mathcal{S}_i\}$ are *accessible* – i.e., can be touched by a extrusion head while not colliding any $\Pi(\mathcal{S}_k)$ $(\forall k < i)$;
3. Every curved layer $\mathcal{S}_i$ is enclosed by the dilation of previous curved layers, $\cup_{k=1,\ldots,i-1}\Pi(\mathcal{S}_k)$, with radius r – i.e., the overhang of $\mathcal{S}_i$ is small

so that an object is self-supported.

Each $\Pi(\mathcal{S}_i)$ represents a solid layer with thickness r that can be fabricated by moving an extrusion head along the surface $\mathcal{S}_i$. We name each surface $\{\mathcal{S}_i\}$ a curved layer, or *working surface*, as the extrusion head will keep moving along it to accumulate solidified material with a thickness r during fabrication. The use of the symbol '$\approx$' above implies a decomposition minimizing the shape approximation error.

We treat the surface decomposition over the whole model under manufacturability condition as a global search problem. However, searching over all possible sequences of surface layers in a continuous volume space is impractical.

To make this amenable to computation, we first discretize space into a regular voxel grid: such a representation is easy to construct, store and process. We then assume that material accumulation during AM is performed by adding voxels one by one. The criteria for validating the feasibility of a manufacturing sequence are converted into geometric constraints between neighboring voxels. The computed sequence of voxel accumulation indicates the flow of fabrication. The sequence is encoded by storing an integer (rank in sequence) at the center of each voxel. This defines a *growing field* $G(\mathbf{x})$.

The efficient computation of a feasible growing field on the voxel grid of $\mathcal{H}$ is presented in Section 3.4. We start by introducing a simple greedy scheme using convex-fronts to ensure the accessibility of the working surfaces in Section 3.4.2. We then introduce the concept of *voxel shadowing* in Section 3.4.3, which is used to avoid the advancing front from a current layer to a next layer to produce inaccessible regions – i.e., regions that become 'behind' the working front and cannot be accessed any more. In Section 3.4.4, we introduce a heuristic based on inverse peeling to control the growth. This strongly reduces the chance of generating shadowed regions, resulting in faster computation and less failure cases.

One drawback of the voxel discretization is to cause severe aliasing of the layer geometries. To solve this problem, we compute the working surface for each layer by extracting a corresponding isosurface $\mathcal{S}^*$ from $G(\mathbf{x})$, as a polygonal surface mesh. A working surface $\mathcal{S}$ with accurate boundary is obtained by trimming $\mathcal{S}^*$ with $\mathcal{H}$ so that the boundary $\partial\mathcal{S}$ of $\mathcal{S}$ is exactly located on the boundary $\partial\mathcal{H}$ of $\mathcal{H}$. Details for extracting a working surface with accurate boundary are presented in Section 3.4.5.

An illustration for this phase of volume-to-surface decomposition in our

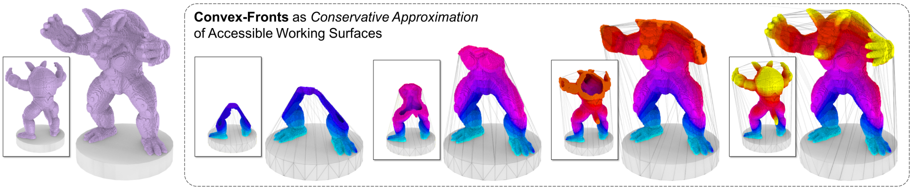


Figure 3.4: Advancing convex-front for collision-free robot-assisted AM. The model is a hollowed Armadillo with 540.6k voxels. Material accumulation is always performed on the *convex-front*: the convex hull of previously deposited voxels and the platform. Back-views are also provided.

framework can be found in Fig.3.3(a)-(c). After obtaining the working surfaces, tool-paths for robot-assisted AM are generated on each of them by solving the surface covering problem below. The resultant tool-paths are illustrated in Fig.3.3(d).

3.3.2 Surface Covering

Given a curved layer surface $\mathcal{S}$ that is feasible, we next consider how to efficiently generate a set of (curved) tool-paths $\{\mathcal{P}_j\}_{j=1,\dots,m}$ such that

1. We cover the layer: $\Pi(\mathcal{S}) \approx \cup_{j=1,\dots,m}\Pi(\mathcal{P}_j)$ with $\Pi(\mathcal{P}_j)$ denoting the convolution solid of $\mathcal{P}_j$ by a sphere with radius r, and there is no overlap between paths – i.e., $\Pi(\mathcal{P}_i) \cap \Pi(\mathcal{P}_j) = \emptyset$ $(\forall j \neq i)$;
2. The number of curves, m, and the distance between the ending points of a tool-path, $\mathcal{P}_j$, and the starting point of the next tool-path, $\mathcal{P}_{j+1}$, are minimized. This reduces the artifacts caused by spurious filaments (so-called *stringing*);
3. The shape of each curve $\mathcal{P}_j$ should be as smooth as possible and be easily realized on a robotic arm.

To meet these conditions while covering the surface, we rely on a variation of Fermat-spiral curves [2], computed on a mesh surface using geodesic distance-fields. The robotic arm introduces additional difficulties, in particular regarding abrupt changes of orientations. Orientations and poses are optimized to realize a robot-assisted AM tool-path compatible with the robotic arm. Details are presented in Section 3.5.

3.4 Decomposition in Curved Layers

This section presents our method for decomposing a given solid model $\mathcal{H}$ into a set of working surfaces $\{\mathcal{S}_i\}_{i=1,\ldots,n}$ for tool-path generation. The problem discretization is first introduced in Section 3.4.1. Then, we present a scheme for generating the growing field $G(\mathbf{x})$ following a greedy strategy (Section 3.4.2), which is later improved by incorporating a mechanism to reduce the apparition of inaccessible regions (Section 3.4.3). A peeling-based heuristic is introduced to further reduce failure cases and to improve performance (Section 3.4.4). Lastly, we describe the extraction of working surfaces from $G(\mathbf{x})$ in Section 3.4.5.

3.4.1 Problem Discretization and Approximation

The input solid model $\mathcal{H}$ is represented by a set of voxels $\{v_{i,j,k}\}$ with a fixed width w (i.e., $\mathcal{H} \approx \bar{\mathcal{H}} = \{v_{i,j,k}\}$), where $\mathbf{c}_{i,j,k}$ denotes the center position of $v_{i,j,k}$ in $\mathbb{R}^3$. $\bar{\mathcal{H}}$ is then converted into a growing field, from which working surfaces are extracted. To allow for the trimming that provides accurate boundaries, the input solid needs to be fully bounded by its voxel representation – that is $\mathcal{H} \subset \bar{\mathcal{H}}$.

We now give the definitions and constraints required for computing feasible sequences of material accumulation.

Definition 1 Two voxels, $v_{i,j,k}$ and $v_{r,s,t}$, are defined as *AM-stable-neighbors* (ASN) to each other if $\|(i,j,k) - (r,s,t)\|_1 \in \{1,2\}$.

Here $\|\cdot\|_1$ denotes the L^1-norm. Note that only face-neighbors and edge-neighbors are considered as AM-stable neighbors: if two voxels are neighboring by only a vertex, the interface between them is deemed too small for reliable accumulation. The ASN set of a voxel $v_{i,j,k}$ is denoted as $\mathcal{N}(v_{i,j,k})$.

Definition 2 A voxel $v_{i,j,k}$ is defined as ϵ-located on a polyhedron $\mathcal{P}$ if the distance, $d(\mathbf{c}_{i,j,k}, \partial\mathcal{P})$, between $\mathbf{c}_{i,j,k}$ and $\partial\mathcal{P}$ is less than ϵ, where $\partial\mathcal{P}$ denotes the boundary of $\mathcal{P}$.

Definition 3 A voxel $v_{i,j,k}$ is defined as *outside* a polyhedron $\mathcal{P}$ if $\mathbf{c}_{i,j,k}$ is outside $\mathcal{P}$ and $d(\mathbf{c}_{i,j,k}, \partial\mathcal{P}) \geq \epsilon$; similarly, $v_{i,j,k}$ is defined as *inside* when $\mathbf{c}_{i,j,k}$ is inside $\mathcal{P}$ and $d(\mathbf{c}_{i,j,k}, \partial\mathcal{P}) \geq \epsilon$.

Material accumulation can be simulated by adding the voxels of $\bar{\mathcal{H}}$ one by one, first onto the manufacturing platform and then onto previously added voxels. While generating the sequence of voxel-additions, the constraints of robot-assisted AM can be applied directly on the growing set of voxels. In particular, two major constraints for manufacturability are considered – *support-free* and *accessibility*.

Constraint 1 (*Support-free*) A voxel can only be accumulated if one of its ASNs has already been solidified (added).

Note that the support-free constraint using ASN allows to accumulate materials along *all* possible directions, reflecting the rotational capabilities of the 6-DOF robotic arm. While this constraint results in a stable accumulation, it however provides no guarantee regarding collisions. This is dealt with through the following constraint.

Constraint 2 (*Accessibility*) When adding a new voxel to a set of already fabricated voxels $\mathcal{V}$, the relative motion of the extrusion head should not collide with $\mathcal{V}$.

This constraint is the most challenging to achieve. It depends on multiple factors, including 1) the size and shape of an extrusion head, 2) the sequence of material accumulation and 3) the local geometry of the working surface. The first factor depends on the hardware. The second and the third factors are coupled with each other, as different sequences result in different working surfaces during fabrication.

When incrementally accumulating materials voxel by voxel, both the conditions of support-free and accessibility have to be verified at all times. The computation of ASN is made very efficient by the voxel-representation. However, collision-detection for accessibility is extremely time-consuming if it is taken explicitly on all voxels. To obtain an efficient planning algorithm, we propose to always ensure that the accumulation is performed on an accessible surface, which can be navigated by the extrusion head without collisions.

The visible surfaces of a model $\mathcal{H}$ are in fact its accessible surface if the tool is infinitely thin, e.g. is a line. In the other extreme case of using a tool with an infinitely large flat head, the accessible surface of $\mathcal{H}$ becomes its convex hull $\mathcal{C}(\mathcal{H})$. Considering that commercial extrusion nozzles have large, nearly flat shapes, the convex hull provides a sensible, conservative approximation of the accessible working surface. Specifically, as the materials are usually accumulated on top of a working platform $\mathcal{T}$, we use the convex hull $\mathcal{C}(\mathcal{V} \cup \mathcal{T})$ as the *conservative* accessible surface for the set $\mathcal{V}$ of voxels that

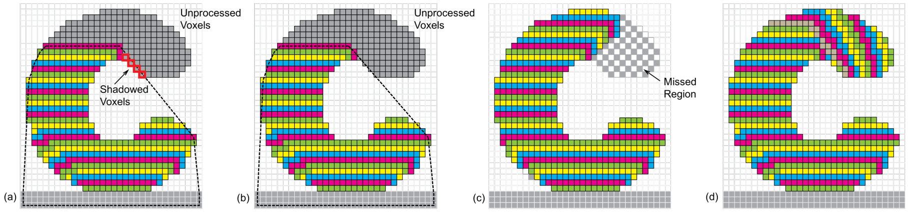


Figure 3.5: An example illustrating shadowed voxel avoidance. Colors are used to indicate voxels belonging to different layers. (a) a critical step where shadowed voxels appear, (b) the shadowed voxels could be avoided by not adding a few voxels to the current layer, (c) the final result without shadow prevention – note the large missing regions – and (d) the result when enabling shadowed voxel avoidance.

have been fabricated. In the remainder of the chapter, this convex hull serves as a progressively enlarged volume-bound to supervise the collision-free motion planning. We call it the *convex-front* (see Fig.3.4 for an illustration of the advancing convex front).

3.4.2 Greedy Scheme for Convex-Front Advancing

As mentioned in Section 3.3, the growing field $G(\cdot)$ is generated by determining an order of voxel accumulation. A voxel v belongs to the l-th layer if $G(\mathbf{c}(v)) = l$ where $G(\mathbf{c}(v))$ is the value of the grid node enclosing $\mathbf{c}(v)$, the center of v. We seek to compute a sequence of feasible layers, $\{\mathcal{L}_l\}_{l=1,\ldots,m}$, where each $\mathcal{L}_l$ consists a set of voxels that meets the support-free and collision-free constraints. Every current layer $\mathcal{L}_c$ is ϵ-located on a convex hull which encloses all prior layers, $\mathcal{L}_l$, with $l < c$. All voxels in $\mathcal{L}_c$ should also be ASN of voxels on the prior layers – i.e., the support-free condition is satisfied for all. Starting from the layer of voxels connected to the platform model $\mathcal{T}$, the greedy scheme generates a sequence of feasible voxel layers. The algorithm opportunistically adds as many voxels as possible into the next layer, following five steps:

1. For a model $\mathcal{H}$ represented by a set of voxels $\bar{\mathcal{H}} = \{v_{i,j,k}\}$, first assign all the voxels attached to the platform $\mathcal{T}$ to the first layer $\mathcal{L}_1$. Set it as the current working layer $\mathcal{L}_c$.
2. All voxels of $\mathcal{L}_c$ are added into the set of *processed* voxels $\mathcal{V}$.
3. The convex hull of $\mathcal{T}$ and all processed voxels in $\mathcal{V}$ is computed as $\mathcal{C}_c = \mathcal{C}(\mathcal{V} \cup \mathcal{T})$. It is the current convex-front.
4. For each voxel $v_{i,j,k} \in \mathcal{L}_c$, any of its unprocessed ASN, $v_{r,s,t}$, (by

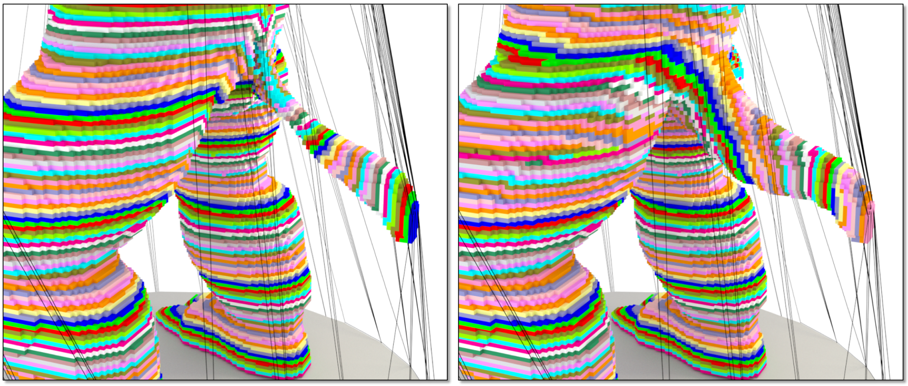


Figure 3.6: The base of the Armadillo model's tail cannot be reached by the greedy scheme of convex-front advancing (left – as a result, 10,752 voxels are missed out of 540.6k voxels). This is improved by considering shadowed regions during growth (right – only 3 voxels are missed). Voxels in the same layer are displayed using the same color.

collision-free condition) will be a candidate voxel to be inserted into the next layer, $\mathcal{L}_{next}$, if $v_{r,s,t}$ is not inside $\mathcal{C}_c$ (Definition 3).

5. If $\mathcal{L}_{next} \neq \emptyset$, set $\mathcal{L}_c = \mathcal{L}_{next}$ and go back to Step (2).

It is easy to find that the most time-consuming step is the computation of $\mathcal{C}_c$ in step (3). To remove the redundancy of computation, the evaluation including all processed voxels can be replaced by only using the previous convex hull $\mathcal{C}_{prev}$ and the newly added voxels in $\mathcal{L}_c$, that is $\mathcal{C}_c = \mathcal{C}(\mathcal{C}_{prev} \cup \mathcal{L}_c \cup \mathcal{T})$. As only the local search and the detection of *in* / *out* convex are included in the computation of this algorithm, it is very efficient – around 32.8 seconds for a model with 540.6k voxels. Pseudo-code for the greedy convex-front advancing is given in **Algorithm** *GreedySchemeCFA*.

This greedy strategy results in curved layers with large areas of connected voxels, that can be later covered with tool-paths. Unfortunately, it also often produces a situation where the convex-front cannot reach all regions of an input model: see the left of Figure 3.6 where the base of the Armadillo tail (with large overhang) cannot be reached. In the following Section, we present an improved strategy that strongly minimizes such failure cases.

Algorithm 1: GreedyGrowingCFA

```
Input: Voxel representation of a solid model, $\bar{\mathcal{H}} = \{v_{i,j,k}\}$
Output: A growing field $G(\cdot)$ with value defined on every voxel of $\bar{\mathcal{H}}$
Adding all voxels adjacent to the platform $\mathcal{T}$ to the first layer, $\mathcal{L}_1$, as a set of voxels;
Set $\mathcal{L}_1$ as the current working layer $\mathcal{L}_c$ and $\mathcal{C}_{prev} = \emptyset$;
Set the layer index $\tau = 1$;
while $\mathcal{L}_c \neq \emptyset$ do
    Add all voxels of $\mathcal{L}_c$ into the already processed set, $\mathcal{V}$;
    Compute the new convex-front by the convex hull of $\mathcal{C}_{prev}$, $\mathcal{L}_c$ and $\mathcal{T}$ as $\mathcal{C}_c = \mathcal{C}(\mathcal{C}_{prev} \cup \mathcal{T} \cup \mathcal{L}_c)$;
    Set $\mathcal{L}_{next} = \emptyset$ and $\tau = \tau + 1$;
    foreach $v_{i,j,k} \in \mathcal{L}_c$ do
        foreach $v_{r,s,t} \in \mathcal{N}(v_{i,j,k})$ do
            if $v_{r,s,t}$ NOT inside $\mathcal{C}_c$ then
                if $v_{r,s,t} \notin \mathcal{V}$ AND $v_{r,s,t} \notin \mathcal{L}_{next}$ then
                    Add $v_{r,s,t}$ into $\mathcal{L}_{next}$;
                end
            end
        end
    end
    foreach $v_{r,s,t} \in \mathcal{L}_{next}$ do
        Assign the field-value as $G(\mathbf{c}(v_{r,s,t})) = \tau$;
    end
    Set $\mathcal{L}_c = \mathcal{L}_{next}$ and $\mathcal{C}_{prev} = \mathcal{C}_c$;
end
```

3.4.3 Preserving Accessibility

We modify the greedy scheme to significantly reduce the apparition of unreachable regions. We call *shadowed voxels* these voxels which can no longer be accessed because they are occluded from the fabrication device by previously fabricated regions.

Definition 4 A voxel, $v_{i,j,k}$ is *shadowed* if it is unprocessed but lies inside the convex hull of the current advancing front, $\mathcal{C}_c$. A set of shadowed voxels form a shadow region, as illustrated in Fig. 3.5.

The fact that a shadowed voxel can no longer be reached is irreversible –

the convex hull is monotonically increasing with respect to the inclusion property during accumulation. Thus our algorithm strives to avoid such cases. This motivates our inclusion of the following constraint to prevent shadowed regions.

Constraint 3 (*Shadow-prevention*) When adding new voxels onto a set of already fabricated voxels $\mathcal{V}$, the number of shadowed voxels should increase as little as possible.

We add a shadow prevention sub-routine before using the next layer of voxels, $\mathcal{L}_{next}$, to update the field values of $G(\cdot)$. The sub-routine selects a reduced set of voxels $\tilde{\mathcal{L}} \subset \mathcal{L}_{next}$ that avoids producing shadowed voxels, as described next.

Incremental Scheme

Our strategy , detailed in Algorithm *IncrementalShadowPrevention*, is to incrementally add voxels from $\mathcal{L}_{next}$ into $\tilde{\mathcal{L}}$, checking them one by one for the apparition of shadowed voxels. Input of the algorithm includes the set of processed voxels $\mathcal{V}$, the next layer from the greedy approach $\mathcal{L}_{next}$, and the current set of shadowed voxels $\mathcal{S}_c$. The algorithm consists of six steps starting from an empty $\tilde{\mathcal{L}}$.

1. Check if any unprocessed voxels are inside
$$\mathcal{C}_p = \mathcal{C}(\mathcal{C}_{prev} \cup \mathcal{T} \cup \mathcal{L}_{next}),$$
and add them into a set $\mathcal{S}_p$ of *potentially* shadowed voxels.
2. If $\mathcal{S}_p \neq \mathcal{S}_c$, generate the reduced set $\tilde{\mathcal{L}}$ by the following steps; otherwise, exit the sub-routine returning $\mathcal{L}_{next}$ (no additional shadowed voxel will be produced).
3. Determine a heuristic sequence $\mathcal{Q}$ to add the voxels from $\mathcal{L}_{next}$ in $\tilde{\mathcal{L}}$.
4. Remove a voxel v from the head of $\mathcal{Q}$, and add v into $\tilde{\mathcal{L}}$ *if* its addition does not increase the set of shadowed voxels when testing with:
$$\mathcal{C}_t = \mathcal{C}(\mathcal{C}_{prev} \cup \mathcal{T} \cup \tilde{\mathcal{L}} \cup v).$$
5. Repeat the above step until $\mathcal{Q} = \emptyset$.
6. If $\tilde{\mathcal{L}} \neq \emptyset$, assign $\mathcal{L}_{next} = \tilde{\mathcal{L}}$. Otherwise, keep the original $\mathcal{L}_{next}$ and update $\mathcal{S}_c$ by $\mathcal{S}_p$ to continue advancing the front, sacrificing the new shadowed voxels in $\mathcal{S}_p$ (i.e., they will not be reached in the future).

Different sequences $\mathcal{Q}$ result in different ‘safe’ subsets $\tilde{\mathcal{L}}$. It is desirable to obtain connected large regions – regions that can be easily covered by

Algorithm 2: IncrementalShadowPrevention

```
   Input: The voxel set of an input model $\bar{\mathcal{H}}$, the set of processed voxels
          $\mathcal{V}$, the next layer $\mathcal{L}_{next}$ and the current set of shadow region $\mathcal{S}_c$
   Output: An reduced set of $\mathcal{L}_{next}$
1  Set $\mathcal{S}_p = \emptyset$, and $\tilde{\mathcal{L}} = \emptyset$;
2  Compute $\mathcal{C}_p = \mathcal{C}(\mathcal{C}_c \cup \mathcal{T} \cup \mathcal{L}_{next})$;
3  $\forall v \in (\bar{\mathcal{H}} \setminus \mathcal{V})$, add $v$ into $\mathcal{S}_p$ if it is inside $\mathcal{C}_p$;
4  if $\mathcal{S}_p = \mathcal{S}_c$ then
5      Return $\mathcal{L}_{next}$;
6  end
7  Determine a heuristic sequence $\mathcal{Q}$ of voxels in $\mathcal{L}_{next}$ by a flooding
     algorithm;
8  while $\mathcal{Q} \neq \emptyset$ do
9      Remove a voxel $v$ from the head of $\mathcal{Q}$;
10     Compute the set of shadowed voxels $\mathcal{S}_t$ according to
         $\mathcal{C}_t = \mathcal{C}(\mathcal{C}_c \cup \mathcal{T} \cup \tilde{\mathcal{L}} \cup v)$;
11     Add $v$ into $\tilde{\mathcal{L}}$ if $\mathcal{S}_t = \mathcal{S}_c$;
12 end
13 if $\tilde{\mathcal{L}} \neq \emptyset$ then
14     Set $\mathcal{L}_{next} = \tilde{\mathcal{L}}$;
15 else
16     Set $\mathcal{S}_c = \mathcal{S}_p$;        // update the set of shadowed
         voxels
17 end
18 return $\mathcal{L}_{next}$;
```

tool-paths. Therefore, starting from a randomly selected source voxel in $\mathcal{L}_{next}$ we use a Dijkstra's algorithm to generate a sequence $\mathcal{Q}$ according to the voxels' distances to the source.

Testing for shadowed voxels in Step (4) potentially requires visiting all remaining unprocessed voxels, which could be extremely slow. However, only the unprocessed voxels within $\mathcal{S}_p$ (determined in Step (1)) can possibly be inside $\mathcal{C}_t$, since $\mathcal{C}_t \subset \mathcal{C}_p$. Thus, the algorithm only has to test a small portion of the unprocessed voxels.

Recursive Shadow-free Sets

The algorithm can be further accelerated. The key idea is to recursively split $\mathcal{L}_{next}$ into subsets until finding those that are safe to add (shadow-free), or until reaching individual voxels producing shadowed voxels.

Specifically, given a subset of voxels $\mathcal{L}_{sub}$ producing shadowed voxels, we divide it into subsets $\mathcal{L}_{sub}^{L}$ and $\mathcal{L}_{sub}^{R}$ by splitting along the longest principal axis obtained from a *Principal Component Analysis* (PCA) of the voxel centers in $\mathcal{L}_{sub}$. The operation is recursed until reaching a shadow-free subset or a subset with only one voxel. The shadow-free subsets are added into $\tilde{\mathcal{L}}$.

This algorithm is faster as we test entire sets of voxels in a single 'shadow' check (the convex hull $\mathcal{C}_t$ is obtained by adding all voxels from the set). Whenever the check is false, all voxels can be added to $\tilde{\mathcal{L}}$ without further testing. Pseudo-code for the overall algorithm can be found in **Algorithm** *AdaptiveRefinementShadowPrevention*.

When running these two algorithms on the Armadillo model shown in Fig.3.4 with $540k$ voxels, $5,455$ minutes is needed for the incremental algorithm while the adaptive refinement algorithm needs only 304 minutes to generate the similar result – i.e., a $17.9\times$ speedup. Similar orders of speedup are observed on other models with smaller number of voxels (e.g., $6.5\times$ on the bunny model in Fig. 3.1 with $97.5k$ voxels).

3.4.4 Inverse Peeling for Guiding the Growth

The algorithm presented so far produces good results, but is slowed down by the many shadow prevention checks. In this section we introduce a heuristic that strongly reduces the need for shadow checks, by guiding the growth towards a good solution.

Our heuristic is motivated by considering the process of material accumulation as an inverse process of material removal in subtractive machining. The basic idea is to construct a material removal order by peeling away voxels from a convex-front of remaining voxels. The peeling process starts from the convex hull of the full object. It then peels aways one (curved) sheet of voxels of constant thickness, and it iterates on what remains. This resembles 5-axis CNC as material is removed along all orientations (i.e., the normal of convex-hull). An illustration of the peeling process can be found in Fig.3.7(a). The order obtained from peeling is stored as an integer rank in each

Algorithm 3: AdaptiveRefinementShadowPrevention

```
Input: The voxel set of an input model $\bar{\mathcal{H}}$, the set of processed voxels
       $\mathcal{V}$, the next layer $\mathcal{L}_{next}$ and the current set of shadow region $\mathcal{S}_c$
Output: An reduced set of $\mathcal{L}_{next}$
Set $\mathcal{S}_p = \emptyset$, and $\tilde{\mathcal{L}} = \emptyset$;
Compute $\mathcal{C}_p = \mathcal{C}(\mathcal{C}_c \cup \mathcal{T} \cup \mathcal{L}_{next})$;
$\forall v \in (\bar{\mathcal{H}} \setminus \mathcal{V})$, add $v$ into $\mathcal{S}_p$ if it is inside $\mathcal{C}_p$;
if $\mathcal{S}_p = \mathcal{S}_c$ then
    Return $\mathcal{L}_{next}$;
end
Call Refinement($\tilde{\mathcal{L}}$,$\mathcal{L}_{next}$);
if $\tilde{\mathcal{L}} \neq \emptyset$ then
    Set $\mathcal{L}_{next} = \tilde{\mathcal{L}}$;
else
    Set $\mathcal{S}_c = \mathcal{S}_p$;  // update the set of shadow points
end
return $\mathcal{L}_{next}$;
/* The recursive function for adaptive
   refinement                                  */
Procedure Refinement ($\tilde{\mathcal{L}}$,$\mathcal{L}_{sub}$)
    Compute the set of shadow points $\mathcal{S}_t$ according to
     $\mathcal{C}_t = \mathcal{C}(\mathcal{C}_c \cup \mathcal{T} \cup \tilde{\mathcal{L}} \cup \mathcal{L}_{sub})$;
    if $\mathcal{S}_t \neq \mathcal{S}_c$ then
        if $|\mathcal{L}_{sub}| > 1$      // Set $\mathcal{L}_{sub}$ has more than one
         voxel
         then
            Divide $\mathcal{L}_{sub}$ into two subset $\mathcal{L}^L_{sub}$ and $\mathcal{L}^R_{sub}$ by PCA;
            Call Refinement($\tilde{\mathcal{L}}$,$\mathcal{L}^L_{sub}$);
            Call Refinement($\tilde{\mathcal{L}}$,$\mathcal{L}^R_{sub}$);
        end
    end
    Add $\mathcal{L}_{sub}$ into $\tilde{\mathcal{L}}$;
    return;
```

voxel, defining a field $F(\cdot)$. Then, the inverse field of $F(\cdot)$ is defined as:

$$\bar{F}(\mathbf{p}) = 1 + (\max_{\forall \mathbf{q} \in \tilde{\mathcal{H}}} (F(\mathbf{q})) - F(\mathbf{p})) \tag{3.1}$$

Algorithm 4: PeelingFieldGeneration

```
Input: Voxel representation of a solid model, $\bar{\mathcal{H}} = \{v_{i,j,k}\}$
Output: An indication-field $\bar{F}(\cdot)$ as the inverse of peeling with value
        defined on every voxel of $\bar{\mathcal{H}}$
Assign all voxels of $\bar{\mathcal{H}}$ into a set of unprocessed voxels, $\mathcal{U}$;
Compute the convex-hull of $\mathcal{U}$ and $\mathcal{T}$ as $\mathcal{C}_c = \mathcal{C}(\mathcal{U} \cup \mathcal{T})$;
Set the peeling index $\tau = 1$;
while $\mathcal{U} \neq \emptyset$ do
    Initialize $\mathcal{P}_c = \emptyset$;
    Assign all voxels that are $\epsilon$-located on $\mathcal{C}_c$ into $\mathcal{P}_c$;
    Update $\mathcal{U}$ as $\mathcal{U} = \mathcal{U} \setminus \mathcal{P}_c$;
    For each $v_{r,s,t} \in \mathcal{P}_c$, assign its field value as $F(\mathbf{c}(v_{r,s,t})) = \tau$;
    Update the convex-front as $\mathcal{C}_c = \mathcal{C}(\mathcal{U} \cup \mathcal{T})$;
    $\tau = \tau + 1$.
end
foreach $v_{i,j,k} \in \bar{\mathcal{H}}$ do
    /* Note that: $\tau$ holds the value of
       $\max_{\forall \mathbf{q} \in \tilde{\mathcal{H}}} (F(\mathbf{q}))$ now.          */
    $\bar{F}(\mathbf{c}(v_{r,s,t})) = 1 + (\tau - F(\mathbf{c}(v_{r,s,t})))$; ;      // By Eq.3.1
end
```

This inverse field is used to guide the previous algorithm, still using shadow prevention. The pseudo-code is given in **Algorithm** *PeelingFieldGeneration*.

Intuitively, this peeling heuristic helps reduce the apparition of shadowed voxels by encouraging the growth to proceed uniformly and progressively towards the outer object surface, curving the layers ahead of time. Without this heuristic, the algorithm grows roughly flat until a ‘downward’ feature is suddenly encountered (see the case in Fig.3.5(a)). The orientation must locally be changed to fill the feature, creating many potential shadowed voxels. On the same case (Fig.3.7(b)) the layers already curve and align with the surface thanks to the peeling order. Thus, fewer voxels are shadowed when the feature is filled. This strongly reduces the difficulty of the shadow checks; for example, the total computing time for the Armadillo in Fig.3.4 is reduced from 304 down to 61 minutes.

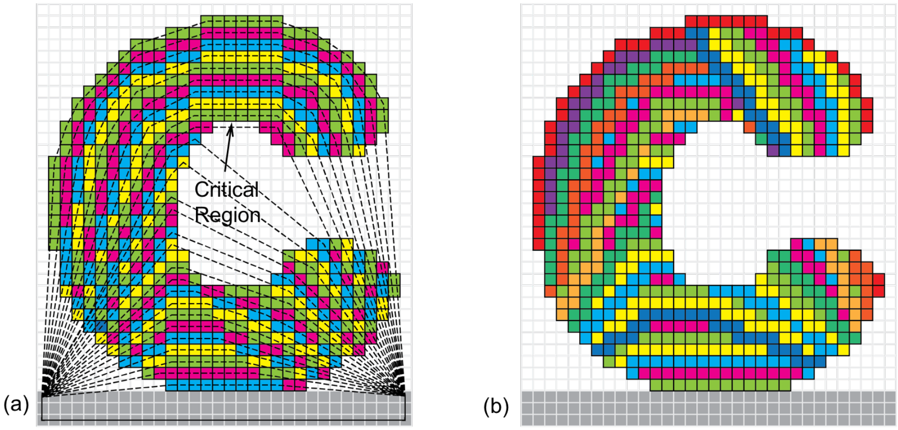


Figure 3.7: An illustration of convex-front peeling and the peeling-governed convex-front advancing. (a) Isolated components will be generated when the voxels in the critical region are removed together with other voxels in the same layer. However, such problem on the peeling field $F(\cdot)$ will be automatically avoided when shadow-region preserved convex-front advancing is conducted – see (b) for a result.

Field-Governed Growing Algorithm

The field $\bar{F}(\cdot)$ generated by inversely peeling provides a very good guidance for the sequence of material accumulation. We revise the growing strategy to follow the field values of $\bar{F}(\cdot)$, controlling the ”speed” of the convex-front growth in different regions. Specifically, we progressively increase a threshold $f_{\mathcal{T}}$ of the field values in $\bar{F}(\cdot)$ and only advance shadow-prevented convex-fronts into regions where the field values $\bar{F}(\cdot)$ are smaller than $f_{\mathcal{T}}$. The algorithm introduces an outer loop above the greedy growing scheme (in Section 3.4.2) to control the speed of material accumulation w.r.t. $\bar{F}(\cdot)$ as follows:

1. Initialize the first layer $\mathcal{L}_1$ by the voxels adjacent to the manufacturing platform $\mathcal{T}$ and the threshold $f_{\mathcal{T}} = \Delta f$.
2. Apply the shadow-prevented greedy convex-front advancing in the set of unprocessed voxels $\{v_{i,j,k}\}$ that satisfy $\bar{F}(\mathbf{c}(v_{i,j,k})) \leq f_{\mathcal{T}}$.
3. Let $f_{\mathcal{T}} = f_{\mathcal{T}} + \Delta f$ and go back to Step (2) until $f_{\mathcal{T}} > \bar{F}_{\max}$ with $\bar{F}_{\max} = \max_{\forall v \in \bar{\mathcal{H}}}(\bar{F}(\mathbf{c}(v)))$.

Note that while advancing the convex-front all the constraints – support-free, accessibility and shadow-free – should be satisfied. However, by using the inverse peeling field $\bar{F}(\cdot)$ as a guiding heuristic, we observe much better performance from the shadow prevention sub-routine. Pseudo-code of the field-governed convex-front advancing is given in **Algorithm** *FieldGovernedCFA*.

3.4.5 Curved Layer Extraction

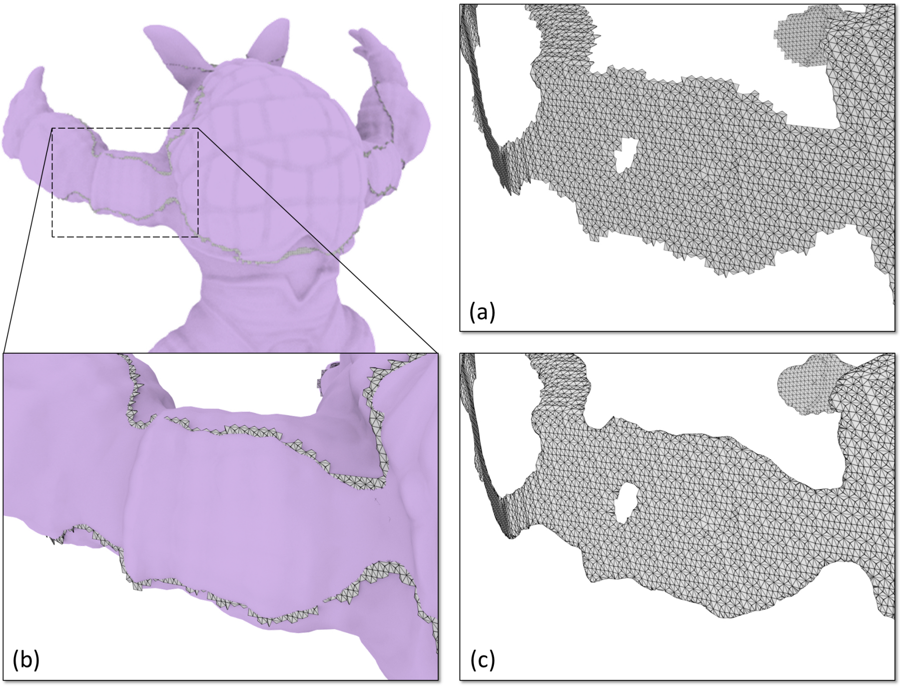


Figure 3.8: An illustration for extracting a working surface with accurate boundary: (a) the isosurface $\mathcal{S}_i^*$ generated by dual contouring, (b) $\mathcal{S}_i^*$ is trimmed by $\mathcal{H}$, to obtain (c) the resultant working surface $\mathcal{S}_i$ with its boundary exactly located on the boundary of $\mathcal{H}$.

Given the growing field $G(\cdot)$ with values defined on every voxel of $\bar{\mathcal{H}}$, working surfaces of curved layers are extracted from $G(\cdot)$ as isosurfaces at different isovalues. Assuming that the size of a voxel is w and the diameter of the extrusion head's nozzle is d, the working surfaces are extracted at the isovalues $i = 1, \ldots, g_i, \ldots, \lceil \max(G(\cdot))/d \rceil$ with:

$$g_i = (i - \frac{1}{2})\frac{w}{d}. \tag{3.2}$$

We first construct a narrow-band grid around the isosurface of $G(\mathbf{p}) = g_i$ by using the voxels which field-values fall within the interval $[\lfloor g_i d/w \rfloor, \lceil g_i d/w \rceil]$.

Algorithm 5: FieldGovernedCFA

```
Input: The voxel representation of a solid model, $\bar{\mathcal{H}} = \{v_{i,j,k}\}$, and
       the governing field, $\bar{F}(\cdot)$.
Output: An indication-field $G(\cdot)$ with value defined on every voxel
        of $\bar{\mathcal{H}}$.
Adding all voxels adjacent to the platform $\mathcal{T}$ to the first layer, $\mathcal{L}_1$, as a
  set of voxels;
Set $\mathcal{L}_1$ as the current working layer $\mathcal{L}_c$ and $\mathcal{C}_{prev} = \emptyset$;
Set the layer index $\tau = 1$ and the threshold as $f_\tau = \Delta f$;
while $f_\tau \leq \bar{F}_{\max}$ do
    while $\mathcal{L}_c \neq \emptyset$ do
        Add all voxels of $\mathcal{L}_c$ into the already processed set, $\mathcal{V}$;
        Compute the new convex-front by the convex hull of $\mathcal{C}_{prev}$, $\mathcal{L}_c$
          and $\mathcal{T}$ as $\mathcal{C}_c = \mathcal{C}(\mathcal{C}_{prev} \cup \mathcal{T} \cup \mathcal{L}_c)$;
        Set $\mathcal{L}_{next} = \emptyset$ and $\tau = \tau + 1$;
        foreach $v_{i,j,k} \in \mathcal{L}_c$ do
            foreach $v_{r,s,t} \in \mathcal{N}(v_{i,j,k})$ do
                if $v_{r,s,t}$ NOT inside $\mathcal{C}_c$ then
                    if $v_{r,s,t} \notin \mathcal{V}$ AND $v_{r,s,t} \notin \mathcal{L}_{next}$ then
                        if $\bar{F}(\mathbf{c}(v_{r,s,t})) \leq f_\tau$ then
                            Add $v_{r,s,t}$ into $\mathcal{L}_{next}$;
                        end
                    end
                end
            end
        end
        Compute the reduced set of $\mathcal{L}_{next}$ for shadow-region
          prevention by Algorithm
          AdaptiveRefinementShadowPrevention;
        foreach $v_{r,s,t} \in \mathcal{L}_{next}$ do
            Assign the field-value as $G(\mathbf{c}(v_{r,s,t})) = \tau$;
        end
        Set $\mathcal{L}_c = \mathcal{L}_{next}$ and $\mathcal{C}_{prev} = \mathcal{C}_c$;
    end
    $f_\tau = f_\tau + \Delta f$;
end
```

We also add their neighboring voxels. Then, a polygonal mesh surface $\mathcal{S}_i^*$ for this isosurface can be extracted by using the *Dual Contouring* (DC) [52] or the *Marching Cubes* (MC) [53] algorithms. In our implementation, we select DC as it generates less polygons. The Hermite information required by DC is obtained by numerical difference on the scalar-field $G(\cdot)$.

Since the boundary $\partial\mathcal{S}_i^*$ of $\mathcal{S}_i^*$ being extracted from a voxel grid, it imperfectly matches the actual boundary of $\bar{\mathcal{H}}$. A surface $\mathcal{S}_i$ with accurate boundary is obtained by trimming $\mathcal{S}_i^*$ with the input polygonal model $\mathcal{H}$ (using, e.g., [54]). This produces a correct result as long as $\partial\mathcal{S}_i^*$ is always *outside* $\mathcal{H}$, which we ensured by using a conservative sampling when constructing the voxel representation $\bar{\mathcal{H}}$ of $\mathcal{H}$. An illustration is given in Fig.3.8.

3.5 Tool-path Planning for Fabrication

Once the geometry of the curved layers is obtained, each has to be covered with tool-paths for material deposition. The basic requirements on the *curved* tool-paths $\{\mathcal{P}_j\}_{j=1,\ldots,m}$ covering a surface $\mathcal{S}$ are: path continuity, orientation continuity and pose continuity.

Our system is similar to FDM printers: filament is heated, melted into viscoelastic material and extruded from an extrusion nozzle through a small hole. This principle makes it difficult to quickly switch extrusion on and off, and therefore a continuous deposition path is demanded. However, position-continuity alone is not sufficient using a robot-assisted AM system. We also have to take into account the variations of orientation. Orientation-continuity is crucial for the fabrication process as it determines the smoothness – and hence quality – of material accumulation. Finally, pose-continuity is necessary to avoid poor dynamic behavior in the motion of the 6-DOF robotic arm. Three requirements are addressed in three steps, using respectively a Fermat spiral curve for continuous tool-paths, a direction optimization for orientation continuity, and a graph-based optimization for pose continuity. We detail each in the following.

3.5.1 Position-continuity

Fused materials in FDM are difficult to control due to the compressibility of molten materials. Existing FDM AM software relieves this problem by generating smooth and continuous tool-paths. A recent effort can be found in [2] to cover a planar domain by a continuous tool-path in Fermat spiral. We

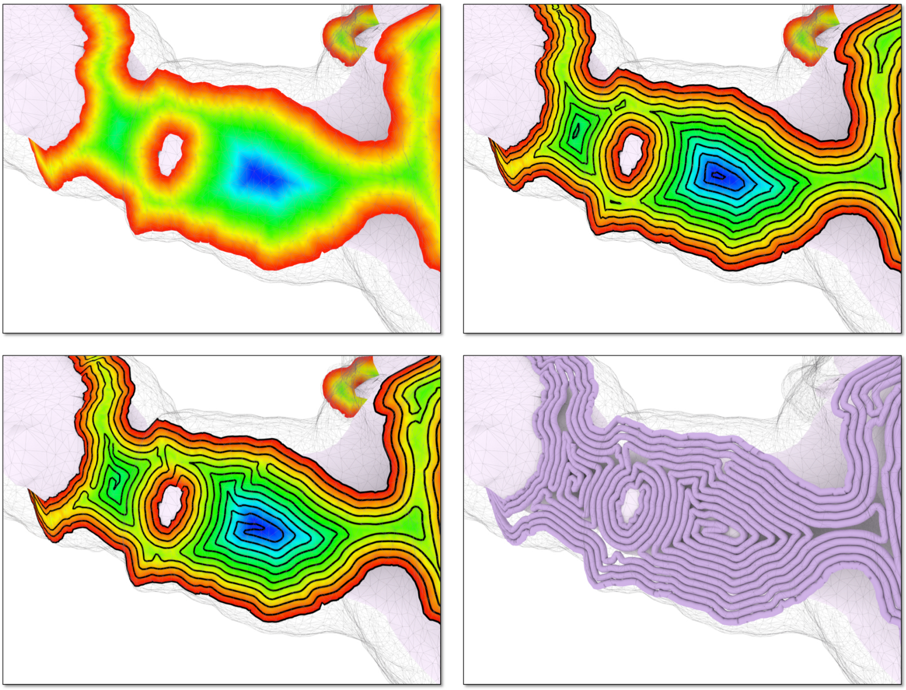


Figure 3.9: An example for illustrating the generation of continuous Fermal spiral tool-path: (Top Row) The geodesic distance field is generated by the FWP-MMP method and the iso-contours are extracted on the mesh surface. (Bottom Row) The iso-contours at different iso-values are connected to form the tool-path.

adopt the same strategy but extend the algorithm to a curved polygonal surface $\mathcal{S}$. The computation in our algorithm relies on a geodesic metric, which is far more difficult to evaluate than the Euclidean distance used in [2]. Our algorithm is applied to each isolated regions of $\mathcal{S}$ separately and consists of three steps:

- First, we build an exact geodesic boundary distance-field over $\mathcal{S}$ by the *Fast-Wavefront-Propagation* (FWP) based on the *Mitchell-Mount-Papadimitriou* (MMP) method [55]. Polyline sources instead of point sources are adopted in order to improve the accuracy of the boundary distance-field.
- Then, we construct iso-contours (i.e., closed-curves having the same iso-value) over the surface mesh $\mathcal{S}$.
- Finally, a 3D Fermat spiral tool-path is generated by connecting iso-contours at different iso-values [2].

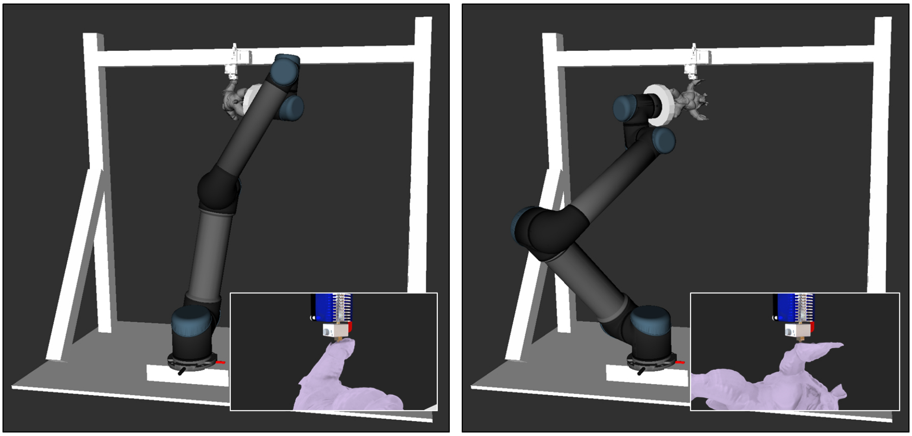
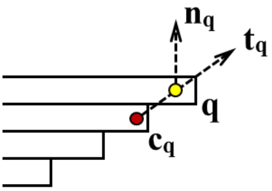


Figure 3.10: A same sample point $\mathbf{c}_j$ with an accumulation orientation $\mathbf{t}_j$ along a tool-path can be realized by the robotic arm using a variety of poses, determined by inverse kinematics. From the closeups, it can be seen that the Armadillo model is actually rotated around the vertical axis of the nozzle between both poses, while the same location is being fabricated.

An example is shown in Fig. 3.9 for the working surface generated in Fig. 3.8.

3.5.2 Orientation-continuity

The Fermat spiral tool-path generated over the mesh surface $\mathcal{S}_i$ consists of many line-segments having different lengths. Each tool-path is uniformly re-sampled into consecutive points with 1mm distance. Then, the orientation of material accumulation at each sample point has to be determined.



The surface normal $\mathbf{n_q}$ at a point $\mathbf{q} \in \mathcal{S}_i$ may seem a natural choice of orientation; however it is not optimized for stable adhesion. We improve the orientation of material accumulation as follows. We first find the closest point $\mathbf{c_q}$ of $\mathbf{q}$ among the surfaces of the curved layers fabricated before $\mathcal{S}_i$:

$$\mathbf{c_q} = \arg\min_{\forall \mathbf{p} \in \mathcal{S}_k} \|\mathbf{p} - \mathbf{q}\|. \quad (\forall k < i)) \tag{3.3}$$

The vector, $\mathbf{t_q} = \mathbf{c_q q}$, provides a better candidate for orientation, as it is consistent with previous layers. However, orientations separately determined on consecutive samples may have large variations. We thus apply a low-pass filter on those samples with orientation-variation larger than $10°$. As a result, orientation continuity can be improved while having better material adhesion.

When the tool-path passes across a crest region, large variation of orientations between two neighboring samples $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ may also be observed. In such cases we subdivide $\mathbf{q}_i\mathbf{q}_{i+1}$ into smaller line segments and compute the orientations for the newly generated sample points by quaternion interpolation. The subdivision significantly improves the dynamic performance of the robotic arm when printing across the crest.

3.5.3 Pose-continuity

Our robot-assisted AM system is built around a 6-DOF robotic arm. Therefore there is an additional DOF available to the arm when moving along the tool-paths with the target orientation. We exploit this additional DOF to optimize the continuity of poses and therefore the dynamic behavior of the robot motion.

Given a list of points with orientations along a tool-path denoted as $\{(\mathbf{c}_j, \mathbf{t}_j)\}$ $(j = 1, \ldots, m)$, we consider the problem of determining corresponding poses of the 6-DOF robotic arm in a joint-angle space. As we are using a hardware setup with a fixed extrusion head, the position and orientation of material accumulation at a point is defined in the frame $\mathbb{B}$ of the end-effector on the robotic arm with the origin located at the center of the working platform.

We first convert $(\mathbf{c}_j, \mathbf{n}_j)$ into p possible poses of $\mathbb{B}$ in the Euclidean space, by rotating $\mathbb{B}$ around the axis of the nozzle (i.e., the z-axis along which to accumulate materials in FDM). In our current implementation, $p = 30$ is used for the sampling rate. This provides a good trade-off between computation time and quality.

For each pose of $\mathbb{B}$, an analytical inverse kinematics solver is applied to determine all possible configurations in the joint-angle space, denoted as $\{\mathbf{a}_{j,k}\}$. A configuration will be excluded when it leads to self-collision or collision with environmental obstacles. As shown in Fig.3.10, poses for realizing a sampling point $(\mathbf{c}_j, \mathbf{t}_j)$ of the tool-path can be significantly different from each other. Therefore, an optimization is taken to generate a sequence of continuous poses $\hat{\mathbf{a}}_j$ that minimizes:

$$\sum_j \|\hat{\mathbf{a}}_j \hat{\mathbf{a}}_{j+1}\|_1 \quad (\exists \hat{\mathbf{a}}_j \in \{\mathbf{a}_{j,k}\}), \tag{3.4}$$

where $\|\cdot\|_1$ denotes a L^1-norm. The problem can be solved on a directed graph by using the Dijkstra's algorithm. Specifically, a node is defined in

$\mathcal{G}_{mo}$ for each pose $\mathbf{a}_{j,k}$. For two neighboring sample points, a directed edge pointing from $\mathbf{a}_{j,k}$ to $\mathbf{a}_{j+1,l}$ $(\forall k, l)$ is constructed with the weight $\|\mathbf{a}_{j,k}\mathbf{a}_{j+1,l}\|_1$ when $\|\mathbf{a}_{j,k}\mathbf{a}_{j+1,l}\|_\infty / \Delta t$ is less than ξ – a hardware-dependent threshold (i.e., $\xi = 1$ rad./sec. is required by our hardware). Here the time step is $\Delta t = \|\mathbf{c}_j \mathbf{c}_{j+1}\| / \bar{v}$ with $\bar{v}$ being a setting linear speed of robot movement in the Euclidean space. Using the multi-source Dijkstra's algorithm (using all nodes of $\mathbf{a}_{1,k}$s as sources), the solution of the problem in Eq.(3.4) can be determined by computing the shortest path linked to one of these sources. $\{\hat{\mathbf{a}}_j\}$ are the nodes of this shortest path, which provides a smooth motion of robotic arm. Note that, it is possible that the graph $\mathcal{G}_{mo}$ is disconnected between $\mathbf{a}_{j,k}$ $(\forall k)$ and $\mathbf{a}_{j+1,l}$ $(\forall l)$ (see Fig.3.11 for an example). In such a case, a smooth motion stops at a pose of $\mathbf{a}_{j,k}$ and thereafter start a new motion from $\mathbf{a}_{j+1,l}$. This disconnection can also be determined by the Dijkstra's algorithm automatically.

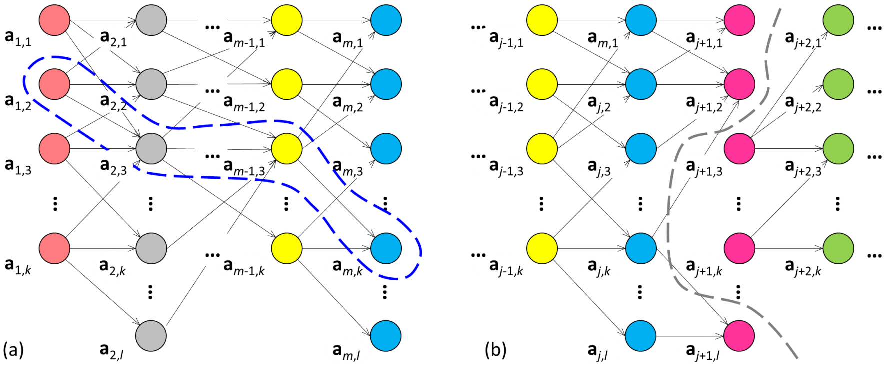


Figure 3.11: Pose-continuity based motion planning is computed on a directed graph: (a) a smooth motion is determined on the graph by the Dijkstra's algorithm of shortest path (circled by the blue dash line) and (b) the disconnection (specified by the gray dash line) between the nodes of two neighboring samples can be found by the same algorithm.

3.6 Results and Discussion

3.6.1 Validation Setup

Physical fabrications using the tool-paths generated by our approach have been taken on a filament-based setup – i.e., *Fused Deposition Modeling* (FDM) equipped with a 6-DOF UR5 robotic arm for multi-axis motion. As shown

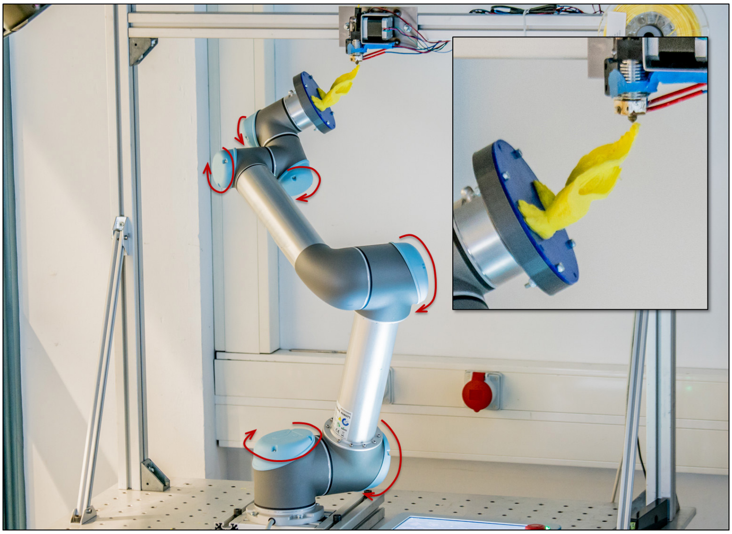


Figure 3.12: The hardware of our robot-assisted AM system using a 6-DOF robotic arm, where each red arrow indicates a motor to provide one rotational DOF. The extrusion head is fixed in our system so that the orientation change for material accumulation is realized by moving the end effect of the robot arm inversely.

in Fig.3.12, we fix the nozzle of FDM extrusion head so that good material adhesion can be obtained comparing to the multi-axis AM with a moving nozzle (e.g., [40, 41]). After installing the extrusion head and the UR5 robotic arm, the relative pose between them is calibrated.

In our hardware system, both the extrusion head and the UR5 robotic arm are controlled by the *robot operating system* (ROS) so that the communication between them can be synchronized. During the fabrication, the required volume of material is determined by the length of a tool-path, and the speed of material extrusion at the nozzle is determined by the time needed for traveling a given path. All are synchronized by ROS.

3.6.2 Experimental Results

We tested our approach on a variety of models. The first example is the hollowed Bunny model shown in Fig. 3.1, discretized in 97.5k voxels. The second and third models are a Candelabra (186.7k voxels) and a hollowed Armadillo (540.7k voxels), shown in Figs. 3.3 and 3.13 respectively. We also tested on models with higher-genus topology: the hollowed Woman-Pully model (185.8k voxels) shown in Fig. 3.13 and the Mech-Part model shown in

Fig. 3.14, which has relatively regular shape but multiple topological handles. Physical printouts are shown in the corresponding figures and Fig. 3.15. As can be seen, our approach successfully exploits the multi-axis motion of the robotic arm to fabricate regions with large overhang without any additional support structures.

Table 3.1: Computational statistics of our approach

		Total	Time (sec.) of Curved Layer Decomposition			Missed	Working Surf.	Tool-Path	Fabrication
Model	Figure	Voxel #	Peeling-Field	Shadow Prev.	Total Time	Voxel #	Time[†] (sec.)	Time (sec.)	Time (min.)
Bunny	3.1	97,532	7.24	171.57	203.90	null	888.72	51.09	119.55
Candelabra	3.3	186,735	9.92	158.83	233.90	null	204.01	131.84	484.81
Armadillo	3.4, 3.13	540,689	34.18	3,035.76	3,639.21	null	1,930.95	567.53	760.11
Woman-Pully	3.13, 3.16	185,815	10.18	436.07	537.06	null	1,258.14	167.75	419.61
Mech-Part	3.14	186,723	n/a	1,200.05	1,252.43	null	444.82	126.72	387.72
Fertility	3.17	77,064	3.79	207.04	232.76	511	-	-	-

[†] The time reported for working surface extraction includes both mesh polygonization and trimming.

Performance data for processing the models are reported in Table 3.1. They are obtained on a DELL desktop with an Intel Xeon E5 1630 3.7GHz Quad Core CPU, 32GB RAM, running Ubuntu 14.04, implemented in C++.

Figure 3.13: The results of our algorithm for maufacturing an Armadillo model and a Woman-Pully model by our robot-assisted AM setup.

We now compare the different strategies we discussed for curved layer decomposition, on both the Armadillo model and the Woman-Pully model. We report in Fig. 3.16 both the computing time and the number of missed voxels. We compare the following strategies:

- *GCFA*: the primary greedy CFA,

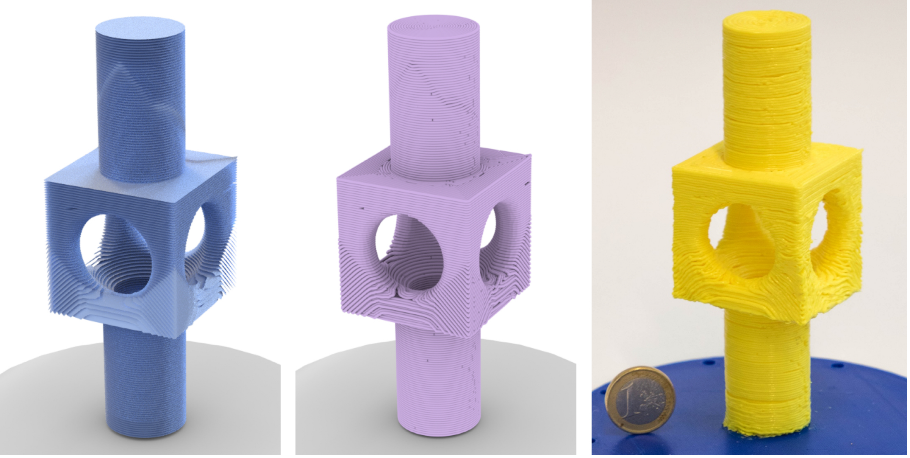


Figure 3.14: The result of our algorithm for a mechanical part – from left to right, the working surfaces, the tool-paths and the physical model.

Figure 3.15: The physical results of all examples shown in this chapter.

- *SP-GCFA*: the shadow-prevented greedy CFA,
- *PG-SP-GCFA*: the peeling-governed and shadow-prevented greedy CFA.

As seen from Fig. 3.16, the PG-SP-GCFA scheme provides the best trade-off

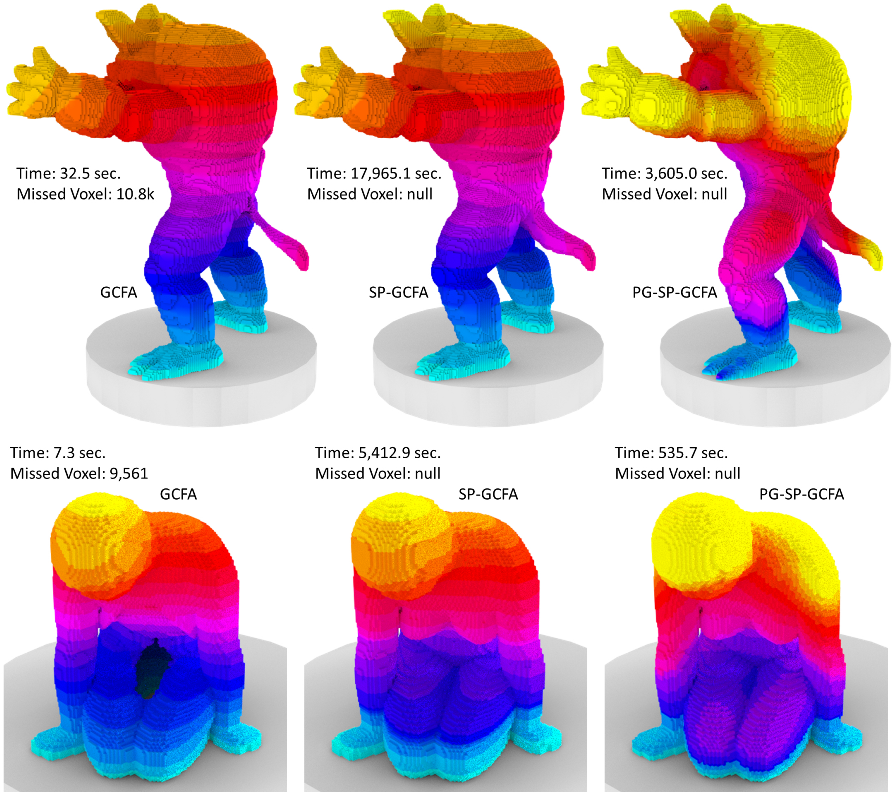


Figure 3.16: The results of different strategies for generating the growth field $G(\cdot)$ on an Armadillo model (with 540k voxels) and a Woman-Pully model (with 185k voxels) – note that both models are hollowed. From left to right, the following schemes are tested: (i) the primary greedy CFA (GCFA), (ii) the shadow-prevented greedy CFA (SP-GCFA) and (iii) the peeling-governed and shadow-prevented greedy CFA (PG-SP-GCFA). The time of computation (in sec.) and the number of missed voxels are also reported. Here the time of PG-SP-GCFA includes the step of generating a peeling field. It can be found that with the help of peeling-governed field the computation of shadow-prevented CFA can be much faster (i.e., 4.98× and 10.1× speedup respectively).

between computation speed and quality. Figure 3.17 shows a failure case of our algorithm on the Fertility model. Neither *SP-GCFA* nor *PG-SP-GCFA* generates a sequence that covers the whole model – i.e., both cannot reach voxels in the chin, although the result of PG-SP-GCFA is obtained much faster and only misses 0.7% of the total voxels. For models like this, a few additional support structures would need to be added for those uncovered regions. See Fig.3.17(c) for the result by adding support to the missed regions.

Since the curved layers are collision–free (convex front), the tool-paths can be directly computed on them without checking for collisions. The tool-path generation for the examples shown in this chapter completes in one to ten minutes. The bottle-neck of our approach is the step checking for shadowed voxels, which can take up to 89.7% of the total time for curved layer decomposition (e.g., the Woman-Pully example). As a result, the computation of curved layer decomposition can take up to hours on some large models. Of course, the time spent on manufacturing a model remains much longer than that of tool-path planning.

3.6.3 Discussion and Limitations

Objects fabricated by our system exhibit artifacts. The main reasons are based on hardware position error, non-uniform layer thickness and gaps between tool-paths, which are discussed below.

Discretization Error

Our approach processes input solids discretized in voxel grids. The aliasing error along the boundary is avoided when extracting curved layers by trimming the extracted iso-surfaces by the original mesh surface (Section 3.4.5). However, the space in-between curved layers varies due to the discrete nature of the growing field. We consider how the actual separation distance between layers differs from the ideal, uniform thickness. Figure 3.18 shows the histogram of distance variations, evaluated by first sampling every curved layer into points and then computing the point-to-surface distances with the PQP library [56]. It is found that the variation of distance is relatively small. In practice, during fabrication the thickness variations are compensated by controlling the feed-rate of material extrusion [57].

Interestingly, the mean of distance variation is not the same as the width of one voxel. This is due to the fact that voxels neighboring by faces and edges are considered in the same way during the front advancing in $G(\cdot)$ – i.e., the field value is increased by one in both cases. As this is a systematic error caused by voxel discretization, the ratio of layer-distance change w.r.t. the width of voxels is expected to remain constant. Experimentally, a ratio of $1.25\times$ appears in most tests – see Fig.3.18. Through this experimental calibration, when a extrusion head with nozzle's diameter d is employed, the computation should be taken on voxels with width $0.8d$. For the Mech-Part example, this ratio can also be used although the resultant layer thickness

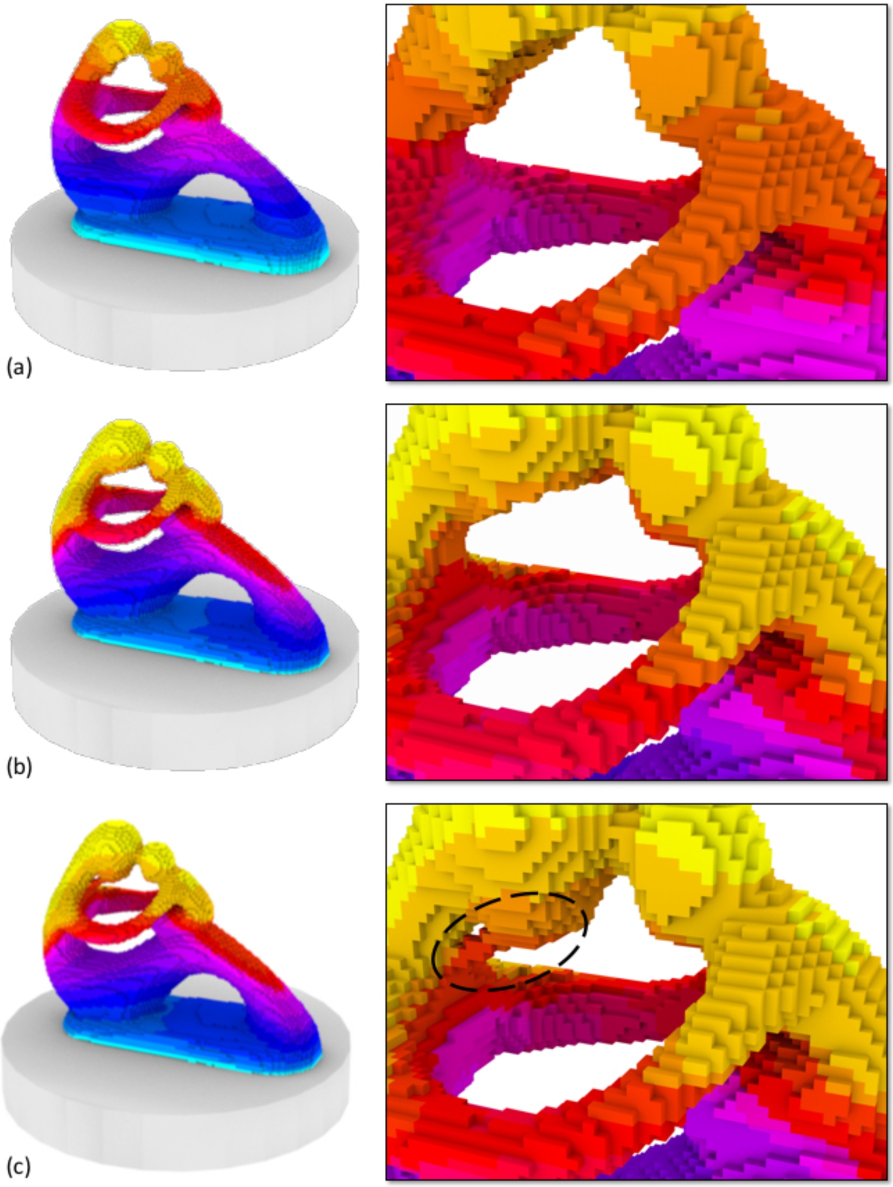


Figure 3.17: An example of failure case – a Fertility model with $77,064$ voxels: (a) the result of SP-GCFA scheme (2,093 seconds with 1,376 voxels missed) and (b) the result of PG-SP-GCFA scheme (142 seconds – 14.7× faster but still have 511 voxels missed). After detecting the missed region and adding supports (encircled by dash line), the growing field can be successfully computed to cover the whole model (c) – PG-SP-GCFA scheme is used here (112 seconds).

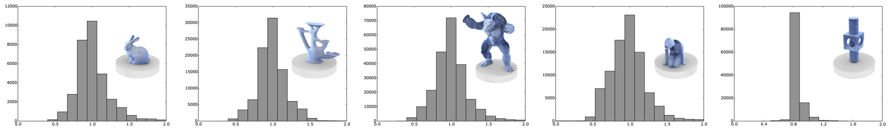


Figure 3.18: The histogram of distance variation between working surfaces on example models shown above (all with voxel width 0.8).

is closer to $0.8d$ instead of d – i.e., the layer thickness $0.8d$ is realized by controlling the feedrate of material extrusion through a nozzle with diameter d.

Hardware

Our approach successfully handles a variety of models including those with high-genus topology and large overhangs; however, we did not explicitly optimize our approach to prevent the generation of thin-features. As a consequence, the quality of material deposition at those thin-features is not very reliable. This is considered as the major limitation on our current FDM-based hardware platform; although this will not be a problem when applying our method on some other platforms (e.g., to fabricate metal parts by arc welding). Another hardware oriented limitation is that the positioning accuracy of the UR5 robotic arm used in our system is relatively low – only with 0.1mm for the repeatability and with around ± 1mm for the positioning error in low speed motion [30], which can be significantly improved when using other high-end systems (e.g., high precision 5-axis table tilting motion system as what is used in 5-axis CNC machining). Figure 3.19 shows a comparison of the Bunny model fabricated on our setup using different tool-paths. It can be found that the artifacts occurs for both results – i.e., positioning inaccuracy on hardware is a major source for manufacturing error.

Our system relies on a fixed extrusion head. As a result, large rotations are applied to the parts during manufacturing, which need to be tightly attached to the end of the robotic arm. Simply using a sticky paper as with the conventional AM method does not work well, as gravity alone can detach the object under fabrication. Our current solution is to first fabricate a working plate using the same material (i.e., PLA in our tests) and fix this plate onto the end of the arm by bolts (see the blue plate in Figs.3.1 and 3.15). The objects are then directly fabricated onto this PLA plate. As the same material is used, the adhesion is strong enough to hold the part. However, the part has to be cut out from the

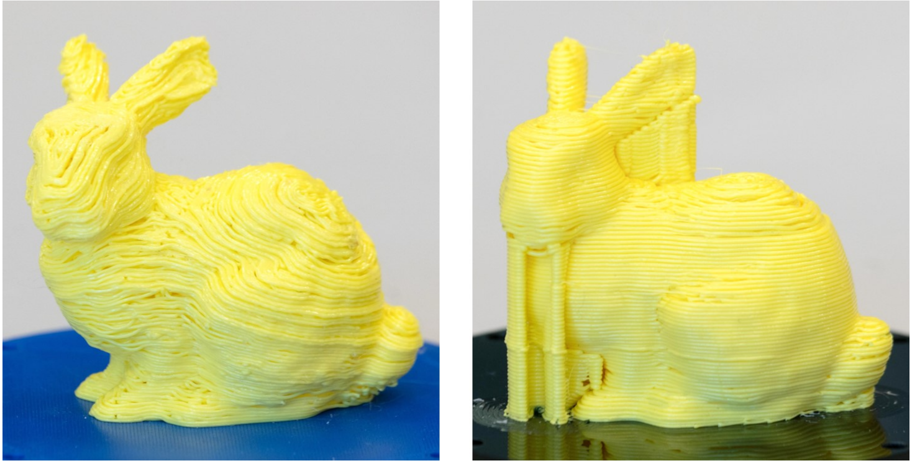


Figure 3.19: A comparison for the Bunny model fabricated on the same robotic system by using different tool-paths: (left) the spatial tool-paths generated by our method and (right) the planar paths generated by conventional slicer for AM.

platform after manufacturing. Optionally the platform could be fabricated in water dissolvable Polyvinyl alcohol(PVA).

Tool-path and Motion Planning

As a problem already identified in [2], Fermat spiral tool-paths show small gaps near the medial axis. Other filling patterns that can result in smooth tool-paths for covering a surface patch more completely should be investigated.

When computing the orientations based on the consideration of best material adhesion (Section 3.5.2), the determined orientation may drive the extrusion-head locally gouging into the already fabricated model. In our current implementation, the orientations are checked and corrected into gouging-free ones locally. A better scheme to compute continuous and optimized gouging-free printing orientations along an input tool-path should be developed. Besides, the current implementation of motion-planning is preliminary – i.e., the dynamic efficiency has not been optimized.

In summary, the capabilities of the support-free robot-assisted AM system are improved in this chapter, thereby enabling the system to handle more complicated geometries compared with the previous work, The next chapter will focus on improving the fabrication quality by planning smoother motion for the robot.

4

Robot-assisted AM: Improving Robotic Motion

This chapter focuses on the robot side of the robot-assisted AM system. It is clear to see that objects fabricated by the previous methods exhibit artifacts. One of the major reasons is that only kinematics of robot is considered in motion planning. The resulting poor dynamic performance brings in the large vibration to the robot's movement, which affects the material deposition and surface quality. Also, the previous motion planning method needs to exhaustively check the collision between the robot and the environment, significantly slowing down the whole fabrication process. To tackle the above limitations, a dedicated motion planning method has been developed in this chapter for generating jerk-optimized robotic motion by considering the kinematic redundancy and using the machine-learning based collision detection. With this method, the printing quality as well as time efficiency have been improved. Several example tool-paths in robot-assisted AM have been tested to demonstrate the functionality of this approach. [3]

4.1 Introduction

In robot-assisted AM system, robotic arms are employed to realize the motion of workpieces (or extrusion head) specified as a sequence of waypoints with the positions of tool tip and the tool orientations constrained (also called tool-paths). The required degree-of-freedom (DOF) is often less than

---

[3] This chapter has been published as: **Chengkai Dai**, Sylvain Lefebvre, Kai-Ming Yu, Jo M.P. Geraedts and Charlie C.L Wang, "Planning Jerk-Optimized Trajectory with Discrete-Time Constraints for Redundant Robots", *IEEE Transactions on Automation Science and Engineering*, accepted, 2020. Note: a few small corrections and/or clarifications have been made to the original published text.

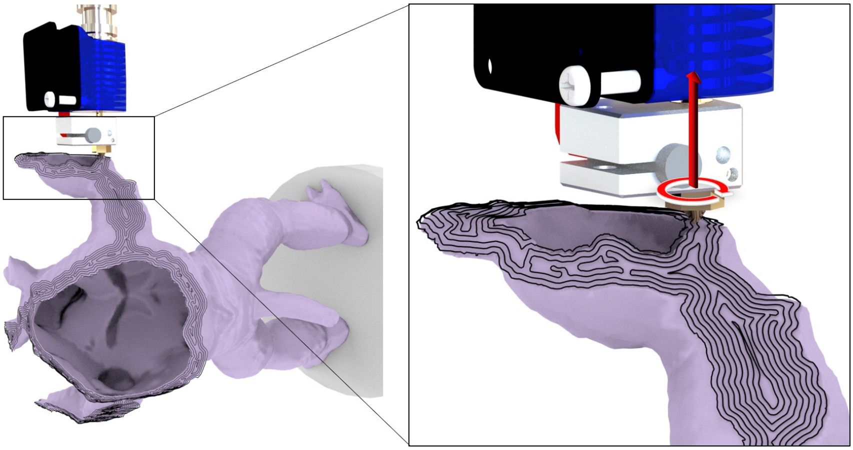


Figure 4.1: An example tool-path for robot-assisted AM system [58], rotation around the red axis can be freely changed because of kinematic redundancy.

the robotic hardware system (e.g., a robotic arm has 6-DOF). Specifically, rotations of the workpiece around the axis of a tool can be arbitrary (see Fig.4.1 for an example). By using this redundancy - i.e., there are many possible poses of a robotic arm to realize a given waypoint, the trajectory of robots can be optimized to consider the performance of motion in velocity, acceleration and jerk in the joint space. In addition, when fabricating complex models each tool-path can have a large amount of waypoints. It is crucial for a motion planning algorithm to compute a smooth and collision-free trajectory of robot to improve fabrication quality. The time taken by the planning algorithm should not significantly lengthen the total manufacturing time; ideally it would remain hidden as computing motions for a layer can be done while the previous layer is printing. The method presented in this chapter provides an efficient framework to tackle this problem. The framework has been well tested on our robot-assisted AM system to demonstrate its effectiveness and can be generally applied to other robot-assisted manufacturing systems.

For realizing the tool-paths for AM, trajectory planning needs to commonly satisfy the following requirements.

- *Discrete-time constraints*: The target trajectories are usually represented as a set of waypoints with given positions and orientations[4] to be accurately passed through at the tip of tool installed on the robot end-effector. Moreover, speed of the tool is also controlled by assigning

---

[4] Orientation is given as a unit vector so that only 5-DOFs are constrained.

a time parameter to each waypoint. Note that speed for material accumulation in AM [17, 59], called *feedrate*, is a very important parameter to be controlled in fabrication.

- *Optimized jerk*: The task of tool-path with discrete-time constraints is mapped into the joint space of a robotic manipulator by its inverse kinematics. As the time derivative of acceleration in joint space, jerk has great influence on the smoothness of a manipulator's motion. A motion with smaller jerk has less vibration. To reduce the vibration into a low level is crucial for realizing a high quality robotic fabrication. An ideal trajectory should have an integral-norm of the jerk minimized and the maximal jerk controlled below a reasonable bound.
- *Collision-free*: Collisions with the surrounding objects and the parts of a model that have already been manufactured must be avoided at all costs along the trajectory of the robot system. While optimizing the jerk, a collision-free property needs to be ensured along the optimized trajectory. This becomes a computational bottleneck when the obstacles have a complex shape. This is unfortunately the case for robotic fabrication as the models to be fabricated usually have complex 3D freeform surfaces.

Generating a trajectory satisfying the aforementioned requirements is challenging. The tool-paths for robotic fabrication can contain a large number of waypoints (typically in the thousands), which are hard to handle with global methods [60]. Online (local) planning methods (e.g., [61, 62]) often include time-jerk optimization strategies to improve the quality of the trajectories. However, they are not applicable to discrete-time constraints. A sampling-based framework is proposed in this chapter to tackle this problem effectively and efficiently.

4.1.1 Problem Statement

Suppose a robotic manipulator with L DOFs ($L > 5$) is employed to follow a user specified path $\mathbf{x}(t) \in \mathbb{R}^5$ with the tool tip held by its end-effector, the location of which is determined by parameters in joint space (i.e., $\mathbf{q}(t) \in \mathbb{R}^L$) by the forward kinematics as:

$$f(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^L \mapsto \mathbf{x} \in \mathbb{R}^5.$$

Given a collision-indication function $\Gamma(\mathbf{q})$ the sign of which indicates a collision-occurring ('$+$') or collision-free ('$-$') configuration, we can define the collision-free configuration space as:

$$\mathcal{C}_{free} = \{\mathbf{q} \mid \Gamma(\mathbf{q}) < 0, \forall \mathbf{q} \in \mathbb{R}^L\}.$$

In robotic fabrication, the problem of finding a feasible jerk-minimized trajectory following the task tool-paths is defined as a complex trajectory planning problem with a large size of input (i.e., many waypoints along freeform surfaces as discrete-time constraints). Given a set of M waypoints each at a prescribed time t_i with the position of tool tip $\mathbf{p}_i$ and the tool orientation $\mathbf{n}_i$, denoted as

$$\mathbf{x}(t_i) = (\mathbf{p}_i, \hat{\mathbf{n}}_i) \qquad (\forall i = 1, \ldots, M), \tag{4.1}$$

we formulate the optimization problem to be solved as

$$\begin{aligned} & \arg\min_{\{\mathbf{q}_i\}} \mathbb{J} = \sum_{i=1}^{M} \|\dddot{\mathbf{q}}(t_i)\|_W^2 \\ s.t. \quad & \mathbf{x}(t_i) = f(\mathbf{q}(t_i)) \quad (\forall i = 1, \ldots, M), \\ & \Gamma(\mathbf{q}(t_i)) < 0, \\ & \mathbf{q}_{\min} \leq \mathbf{q}(t_i) \leq \mathbf{q}_{\max}, \\ & |\dot{\mathbf{q}}(t_i)| \leq \mathbf{v}_{\max},\ |\ddot{\mathbf{q}}(t_i)| \leq \mathbf{a}_{\max},\ |\dddot{\mathbf{q}}(t_i)| \leq \mathbf{j}_{\max}. \end{aligned} \tag{4.2}$$

Here the measure of jerk is $\|\dddot{\mathbf{q}}\|_W^2 = \dddot{\mathbf{q}}^T \mathbf{W} \dddot{\mathbf{q}}$ with $\mathbf{W}$ being a non-negative diagonal matrix giving the weights for relative importance between the joints which can also be assigned as equal importance by $\mathbf{W} = \mathbf{I}$, and $| \cdot |$ returns a vector with the absolute value of every component. The last four constraints (i.e., the last two lines in Eq.(4.2)) are about joint's position, velocity, acceleration and jerk, and are defined according to the hardware limits. $\hat{\mathbf{n}}_i$ is a normalized vector so that only 5-DOFs are constrained. For the sake of compact notation, we denote $\mathbf{x}(t_i)$ and $\mathbf{q}(t_i)$ as $\mathbf{x}_i$ and $\mathbf{q}_i$ in the rest of this chapter.

Note that the quality and feasibility of a trajectory is evaluated at discrete-time samples in our formulation. We argue that a weak form solution (i.e., resolution completeness) for the jerk-optimized trajectory planning is obtained when the sampling points are dense enough. Similar strategies have been used and adopted by the robotics community for motion planning [63, 64].

4.1.2 Our Approach

Directly solving the jerk optimization problem along a trajectory with discrete-time constraints is time-consuming even if the state-of-the-art method such as [60] is employed. In a relaxed formulation, we propose a greedy algorithm based on *local adaptive filtering* to the jerk after computing an initial trajectory that minimizes the total velocity variation. We overcome the computational bottleneck of collision detection by a learning-based collision estimator that approximates the continuous decision function.

- An adaptive greedy algorithm to generate jerk-optimized trajectory with discrete-time constraints (Section 4.3);
- An adaptive sampling strategy for effectively learning a collision-indication function (Section 4.4).

In summary, we develop a new sampling-based framework for planning discrete-time constrained trajectory on redundant robots, which can effectively and efficiently generate jerk-optimized trajectories for robotic fabrication.

4.2 Related Work

In this section, we review the prior research related to the two major parts of our framework: the trajectory planning approaches considering the task-oriented constraints and the machine-learning based collision-detection approaches.

4.2.1 Trajectory Planning for Task-Oriented Constrains

Many robot-assisted manufacturing tasks impose constraints on the robot's motion that exhibit kinematic redundancy, where more DOFs are available than the needed DOFs to realize the task. An example is to accumulate materials in AM process by tracing a given path with an axi-symmetric filament extruder [58, 44]. In robotic motion planning, there are two types of redundancy, intrinsic and functional ones. Intrinsic redundancy occurs when the dimension of the joint space spanned by a robot's joint variables is greater than the dimension of its operational space, which is the reachable Cartesian space of the end-effector. Functional redundancy is the case where the dimension of the robot's operational space is greater than the dimension of the task space (e.g., the waypoints to be realized).

The intrinsic redundancy problem has been discussed extensively by robotic researchers. Most of the existing methods are playing with the null space of the Jacobian matrix, since the Jacobian matrix is non-square, and exploit the self-motion space of redundant robots [65, 66]. Functional redundancy is different from intrinsic redundancy, in which the Jacobian matrix is non-singular and square. It is not possible to obtain the null space of the robot itself. Huo and Bason [67] proposed to add an extra column to the Jacobian matrix by introducing a virtual joint. After that, a general solution called *twist decomposition algorithm* was introduced by using the projection matrices in the operational space to find the null space of Jacobian.

Existing generic trajectory planning methods approach functional redundancy by local [61, 62] or global [68] optimization techniques. The optimization is usually based on different objective metrics such as avoiding obstacles [69], avoiding joint limits [67], avoiding singularities [70] and/or minimizing joint velocities [71], jerks [72] and torques. Some approaches (e.g., [67, 73]) optimize the combination of multiple criteria.

- In local optimization, the strategy is to generate joint configurations [61, 62] that minimize the instantaneous value of the aforementioned metrics. However, these optimization methods only guarantee that a local minima of the objective function is found, which may not be sufficient to ensure path continuity as a whole.
- Global optimization seeks to generate trajectories that minimize the integral of the performance metric over a prescribed interval, as opposed to just instantaneously in time. However, these approaches are time-consuming because that the geometric algorithm for obstacle avoidance has been included in the loop of computation (ref. [60, 73]). The optimizations also suffer from the initial guess problem i.e., whether solutions can be found highly relies on the initial guess. Thus, existing global approaches cannot guarantee an algorithmic completeness (some inputs may not lead to solutions).

Differently, our approach is sampling-based and can ensure algorithmic completeness.

In the area of motion planning, sampling-based algorithms are the most successful method because of their efficiency and completeness [74, 75]. To deal with the redundancy, researchers define constraints as manifolds and efficiently sample the manifolds by forcing the constraints via rejection sampling and projection sampling of the configuration space [76, 77]. However, such algorithms cannot handle the tracking problem with a predefined path. Our work is partially similar with a new global graph search method [78], where the self-motion space is parameterized by angular and path distances and a graph is constructed by cell-decomposition applied to these two parameters. The trajectory planning problem is solved by a shortest path search on the graph. However, the method can be very slow when a dense sampling is applied. In addition, it is unclear how to optimize for minimal jerkiness by such a graph-search based method.

4.2.2 Machine-Learning Based Collision Detection

When performing the trajectory planning in the configuration space, the solutions falling in the sub-space of collision-occurring configurations should be abandoned. For realizing the collision avoidance, the collision-free configuration space should be computed and represented in an appropriate way. As discussed in [79], geometry-based methods are usually limited to low-dimensional configuration spaces, due to the combinatorial complexity involved in computing the boundary of the collision-free space for high-dimensional configurations. Moreover, for computing the optimal trajectory in a numerical optimization framework, the collision-free configuration space is desired to be presented (or approximated) by an algebraic function (i.e., $\Gamma(\cdot)$ as discussed above in Eq.(4.2)).

To solve this problem, machine learning techniques have been used for collision detection to approximate collision-free spaces based on sampled configurations. For example, Pan et al. [80] conducted the incremental *support vector machines* (SVM) to learn a representation of configuration space in an online step. Their method samples the configuration space by iteratively exploiting the near boundary configurations. Das et al. [81] developed a kernel-based perceptron learning algorithm which can efficiently update the classifier actively. This enables the function to online update the decision boundary of a classifier. *Gaussian mixture models* (GMM) are used in [82] to represent the collision-aware configuration space, from which the collision detection is performed by assigning a query configuration with the same label as the closest Gaussian. Pan and Manocha [83] adopt a *k-nearest neighbor* (k-NN) model in their sampling-based motion planners, which can significantly reduce the time required for collision checking. Neural networks also have been applied to perform collision checking (e.g., [84]); however, the training step could be time-consuming when multiple-layers need to be trained in a neural network. Recently, Salehian et al. [85] develop an exhaustive sampling method to find a collision decision function, which can be treated as collision constraints in the optimization-based computation of inverse kinematics. Although potentially applicable after certain modifications, none of the above approaches have considered the specific situation in robot-assisted manufacturing where the shape of obstacles (specifically the workpiece to work on) is changing with time. For instance in AM process the part being created is making the space of collision-free configurations increasingly complex. In this chapter, we develop a sophisticated method to tackle this manufacturing-oriented situation, which can use a very small number of training samples to provide an accurate function for efficiently estimating the collision-indication function in the numerical optimization framework.

4.3 Trajectory Planning

Given a set of waypoints as discrete-time constraints for robotic fabrication, our trajectory planning algorithm computes the configurations of manipulators as a sequence of optimized configurations $\mathcal{Q} = \{\mathbf{q}_i\}$ $(i = 1, \ldots, M)$ in two steps. Firstly, a trajectory is determined by a graph-search method that minimizes the total cost of joint transition – i.e., the initial values of $\mathbf{q}_i$s are assigned. In the second step, a greedy algorithm is developed to optimize the trajectory by locally applying adaptive filters to adjust the value of $\mathbf{q}_i$s in regions with large jerks. A validation mechanism is developed to ensure the resultant trajectory is completely collision-free at the waypoints.

4.3.1 Initialization

We employ a sampling-based method to determine a feasible trajectory. When the sampling rate is dense enough, it provides a very good solution for determining an initial trajectory to be further optimized.

Since only position and orientation are defined on a waypoint, the tool is allowed to rotate freely around the tool axis orientation $\hat{\mathbf{n}}_i$. Therefore, this leads to infinite possibilities to define a pose in the robotic manipulator's configuration space, which results in kinematic redundancy. The rotation can be defined by a quaternion as a rotational angle θ around the vector $\hat{\mathbf{n}}_i$, that is

$$\mathbf{h}_i = (\hat{\mathbf{n}}_i \sin(\frac{\theta}{2}), \cos(\frac{\theta}{2})) \quad (\forall \theta \in [-\pi, \pi]). \tag{4.3}$$

We first sample the task space at every waypoint by using different values of θ, and then employ a graph-search approach to find a feasible path by connecting the selected samples – one from each waypoint.

A graph $\mathcal{G}$ spanning the task space of a given path $\mathcal{X} = \{\mathbf{x}_i\}$ is constructed by the following method (see also Fig. 4.2 for an illustration).

- *Nodes*: Each waypoint $\mathbf{x}_i = (\mathbf{p}_i, \hat{\mathbf{n}}_i)$ is first uniformly sampled into n rotational angles for $\theta \in [-\pi, \pi]$ to determine n quaternions by Eq.(4.3). This leads to n points in the special Euclidean group $\mathbb{SE}(3)$. For each point in $\mathbb{SE}(3)$ multiple kinematic solutions in the joint space can be determined by *inverse kinematics* (IK). Without loss of generality, we assume that N_i points ($N_i > n$) in the joint space can be obtained for realizing a waypoint $\mathbf{x}_i$ – denoted by $\mathbf{q}_{i,j}$ $(j = 1, \ldots, N_i)$. They are defined as a ladder of nodes, $\mathcal{G}_i = \{\mathbf{q}_{i,j}\}$, in $\mathcal{G}$. Each ladder of nodes is displayed as a column of nodes in Fig. 4.2. Nodes corresponding to a configuration with collision are excluded from the graph, which can be efficiently checked by a collision detection library (e.g., [51]).
- *Edges*: Directed edges are constructed by linking nodes in a ladder $\mathcal{G}_i$ to nodes in the next ladder $\mathcal{G}_{i+1}$ while respecting the joint velocity limits. The edge between $\mathbf{q}_{i,j}$ and $\mathbf{q}_{i+1,k}$ is only added when

  $$\frac{\mathbf{q}_{i+1,k} - \mathbf{q}_{i,j}}{t_{i+1} - t_i} \leq \mathbf{v}_{\max}. \tag{4.4}$$

  To avoid the 'winding effect' caused by revolute joints, we evaluate the circular distance between configurations here and also the rest of this chapter. The following transition cost is added as the weight of an edge (angular-velocity estimation):

  $$w(\mathbf{q}_{i,j}, \mathbf{q}_{i+1,k}) = (\mathbf{q}_{i+1,k} - \mathbf{q}_{i,j})^T \mathbf{W} (\mathbf{q}_{i+1,k} - \mathbf{q}_{i,j}) \tag{4.5}$$

  with $\mathbf{W}$ being the non-negative diagonal matrix that gives the weights of relative importance between the joints.

A shortest path on $\mathcal{G}$ from a start node $\mathbf{q}_s \in \mathcal{G}_1$ to an end node $\mathbf{q}_e \in \mathcal{G}_M$ actually defines a trajectory $\mathcal{Q}$ that minimizes the total cost of transition as follows

$$\mathbb{J}_{trans} = \sum_{i=1}^{M-1} \|\mathbf{q}_{i+1} - \mathbf{q}_i\|_W^2. \tag{4.6}$$

The shortest path $\mathcal{P}$ on $\mathcal{G}$ can be found by applying the Dijkstra's algorithm for multiple sources – using all nodes in $\mathcal{G}_1$ as the sources. After updating costs on all nodes in $\mathcal{G}$, the shortest path can be traced back from a node in $\mathcal{G}_M$ having the smallest cost. We start from the sampling rate with $n = 4$. If no path can be found on $\mathcal{G}$, we double the value of n to generate a graph with denser nodes and search the path again.

A shortest path that minimizes the total cost of transition $\mathbb{J}_{trans}$ does not directly lead to an optimized path with minimum jerk. However, it provides

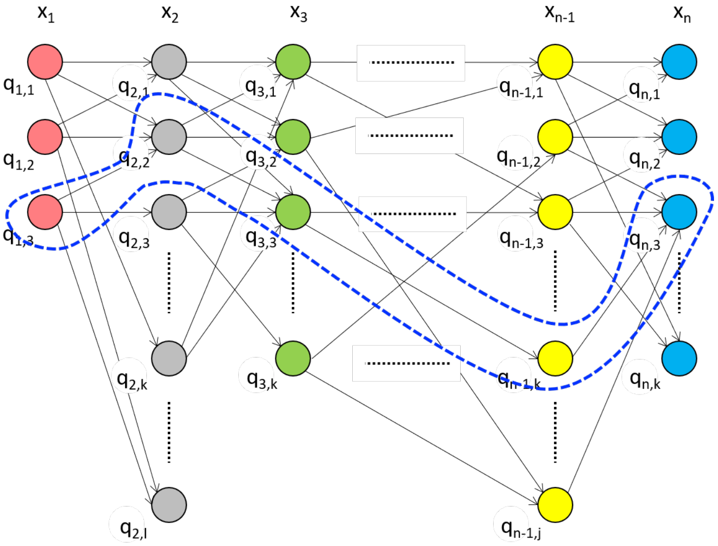


Figure 4.2: An illustration for the graph used in our approach to find an initial trajectory. Nodes in the same column (called ladder) represent the different feasible solutions in the joint space for realizing the same way point. Edges are added between nodes in neighboring ladders. The shortest path on the graph is highlighted by the blue dashed lines.

a good initial path to be further optimized. This algorithm for obtaining an initial trajectory is resolution complete. It means that the algorithm in finite time either finds a solution if one exists, or correctly reports failure. The failure case happens when there exists no path satisfying the velocity limit $\mathbf{v}_{\max}$.

4.3.2 Improvement by Local Filtering

With a given sequence of configurations $\mathcal{Q}$ in joint space according to the waypoints in $\mathcal{X}$, the jerks of trajectory at these discrete-time points can be evaluated by the method of local approximation – the formulation will be given below. In general, the trajectory determined by computing the shortest path on the graph $\mathcal{G}$ in the initialization step may also lead to jerky motion as the initial path does not directly minimize the total jerk (i.e., $\mathbb{J}$ in Eq.(4.2)). To reduce the total jerk on a trajectory with large number of waypoints, we develop an algorithm to improve the trajectory in an iterative routine after obtaining an initial trajectory by the graph-search method presented above.

Our algorithm is based on a greedy strategy. The pseudo-code of our

Algorithm 6: Jerk Optimization by Local Filtering

```
Input: An initial joint trajectory $\mathcal{P}$.
Output: An optimized trajectory $\mathcal{P}$.
Set all points in $\mathcal{P}$ as free;
Find a point $\mathbf{q}_c$ with maximal jerk on $\mathcal{P}$;
while $|\dddot{\mathbf{q}}_c| > \mathbf{j}_{\max}$ AND not enough iterations do
    Build a local path $\tilde{\mathcal{P}}$ centered at $\mathbf{q}_c$ with margin $d$;
    $success = false$;
    while $success \neq true$ AND $d \leq d_{\max}$ do
        $success$ = Minimize $(\mathbb{J}_{local}(\tilde{\mathcal{P}}))$;
        if $success$ then
            Mark all points in $\tilde{\mathcal{P}}$ as free;
        end
        else
            Expand $\tilde{\mathcal{P}}$ centered at $\mathbf{q}_c$ by $d = d + 5$;
        end
    end
    if $success \neq true$ then
        Mark all points in $\tilde{\mathcal{P}}$ as locked;
    end
    Find $\mathbf{q}_c$ with maximal jerk only among the free points on $\mathcal{P}$;
end
return $\mathcal{P}$;
```

algorithm is given in Algorithm 6. At each iteration, we choose a trajectory point $\mathbf{q}_c$ with the maximum jerk among all points as $c = \arg\max_i\{\|\dddot{\mathbf{q}}_i\|_W\}$. A local path $\tilde{\mathcal{P}}$ is extracted with $\mathbf{q}_c$ as the center by using a margin d (i.e., $\tilde{\mathcal{P}} = \{\mathbf{q}_a, \cdots, \mathbf{q}_b\}$ with $b, a = c \pm d$). We apply the local filtering to optimize the local path $\tilde{\mathcal{P}}$ at the jerk level by solving the total-jerk problem defined on this local region (as $\mathbb{J}_{local}$ in Eq.(4.12)). The following algorithm is developed to further enhance the capability in jerk optimization.

- *Window-Size Adaptation*: When no feasible solution is found – i.e., the value of $\mathbb{J}_{local}$ cannot be reduced, it means the computation of optimization is stuck at a local minimum. We then enlarge the window-size of filtering by including more trajectory points until it reaches a user-specified bound.
- *Locking Mechanism*: A locking mechanism is developed in our algorithm to further enhance its capability to overcome the local

> optimum. Specifically, when the window-size of a local path has reached its maximally allowed bound, we will mark all the points in this path as *locked*. The locked points will not be included in the further selection of points with maximal jerk (i.e., the center of local path extraction). A point will be *unlocked* if it has been covered by some other local path, the locally defined total-jerk of which can be reduced.

The adaptive local filtering with locking mechanism is repeatedly applied to regions with maximal jerk until the trajectory meets the required jerk-limit $\mathbf{j}_{\max}$ or the maximum number of iteration is reached (which however rarely occurs in our experiments). Note that, the requirement on maximal jerk is achieved by the algorithm instead of the numerical optimization conducted during the local filtering.

Details for evaluating derivatives at waypoints and computing local-filter at the jerk level are presented as follows.

Derivatives at Waypoints

To compute the derivatives of $\mathbf{q}$, we construct local curves interpolating the waypoints. Specifically, we have

$$\mathbf{q}_i(t) = \sum_{j=-2}^{2} B_j(t)\mathbf{q}_{i+j} \tag{4.7}$$

with the basis functions $B_j(t)$ determined by imposing the interpolation constraints: $\mathbf{q}_i(t_{i+j}) = \mathbf{q}_{i+j}$ ($\forall j = -2, \cdots, +2$). This results in

$$B_j(t) = \sum_{k=0}^{4} b_{k,j+2} t^k, \tag{4.8}$$

where $b_{k,j+2} = \beta_{k+1,j+3}$ with

$$[\beta_{a,b}]_{5\times 5} = \begin{bmatrix} 1 & t_{i-2} & t_{i-2}^2 & \cdots & t_{i-2}^4 \\ 1 & t_{i-1} & t_{i-1}^2 & \cdots & t_{i-1}^4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_{i+2} & t_{i+2}^2 & \cdots & t_{i+2}^4 \end{bmatrix}^{-1}. \tag{4.9}$$

As a result, $\dddot{\mathbf{q}}(t_i)$ can be approximated as:

$$\dddot{\mathbf{q}}(t_i) = \sum_{j=-2}^{2} \dddot{B}_j(t_i)\mathbf{q}_{i+j}. \tag{4.10}$$

When having a constant time-interval $h = t_{i+1} - t_i$ between all waypoints and letting $t_i = 0$, we will have a very compact formula in this special case:

$$\dddot{\mathbf{q}}(t_i) = \frac{\mathbf{q}_{i+2} - 2\mathbf{q}_{i+1} + 2\mathbf{q}_{i-1} - \mathbf{q}_{i-2}}{2h^3}, \tag{4.11}$$

This is in fact the central finite difference formula for the third order numerical derivative obtained by applying the Taylor expansion.

Filter at the Jerk Level

Our filter is applied to a sequence of joint configurations as $\{\mathbf{q}_a, \mathbf{q}_{a+1}, \cdots, \mathbf{q}_b\}$ to minimize the jerk while still satisfying the discrete-time constraints at $\{\mathbf{x}_a, \mathbf{x}_{a+1}, \cdots, \mathbf{x}_b\}$. We formulate the filter as a local optimization problem to minimize the sum of jerks at the waypoints.

$$\begin{aligned} &\min_{\{\mathbf{q}_a, \cdots, \mathbf{q}_b\}} \mathbb{J}_{local} = \sum_{i=a}^{b} \|\dddot{\mathbf{q}}(t_i)\|_W^2 \\ s.t.\quad &\mathbf{x}(t_i) = f(\mathbf{q}(t_i)) \quad (\forall i = a, \ldots, b), \\ &\Gamma(\mathbf{q}(t_i)) < 0, \\ &\mathbf{q}_{\min} \leq \mathbf{q}(t_i) \leq \mathbf{q}_{\max}, \\ &|\dot{\mathbf{q}}(t_i)| \leq \mathbf{v}_{\max}, \ |\ddot{\mathbf{q}}(t_i)| \leq \mathbf{a}_{\max}. \end{aligned} \tag{4.12}$$

The optimization problem defined in Eq.(4.12) is a non-convex problem with non-linear constraints. We therefore use *sequential quadratic programming* (SQP) to solve it. When evaluating $\dddot{\mathbf{q}}_a$ and $\dddot{\mathbf{q}}_b$, those waypoints located at the margin (i.e., $\dddot{\mathbf{q}}_{a-2}$, $\dddot{\mathbf{q}}_{a-1}$, $\dddot{\mathbf{q}}_{b+1}$ and $\dddot{\mathbf{q}}_{b+2}$) are involved as constants instead of variables. Therefore, when applying this local filter to different regions of the path $\mathcal{P}$, we should reserve a margin with at least *four* points between the regions that will be locally updated. Note that we do not impose the requirement of maximally allowed jerk $\mathbf{j}_{\max}$ in this local filter as it is considered at the algorithm level.

The maximally allowed number of iterations in our greedy algorithm is set as 100 in the implementation. Other parameters are set as $d = 5$ and $d_{\max} = 20$ according to our experiments. As shown in Figs. 4.3 and 4.4, our algorithm for trajectory optimization can effectively and efficiently reduce both the total jerk $\mathbb{J}$ and the maximal jerk on initial trajectories determined by the graph search.

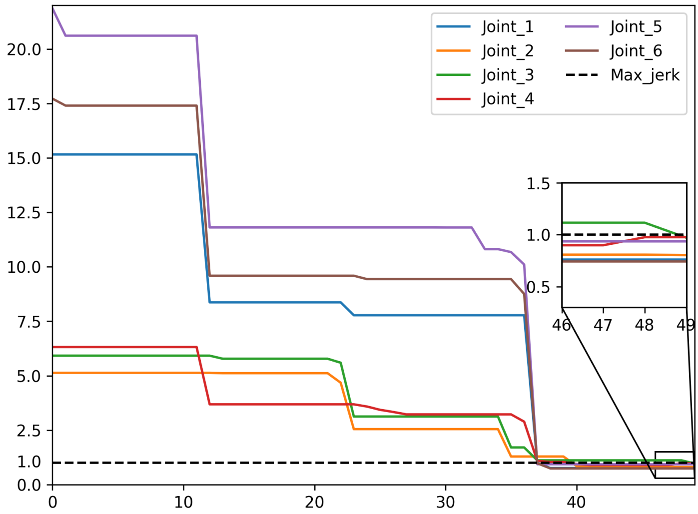


Figure 4.3: The change of the maximum jerk at each joint during the iterations of our method. The maximal jerk has been reduced by $83.6\% - 95.8\%$ on all the six joints. The dash line shows the allowed maximal jerk as $j_{\max} = 1.0$ in this example.

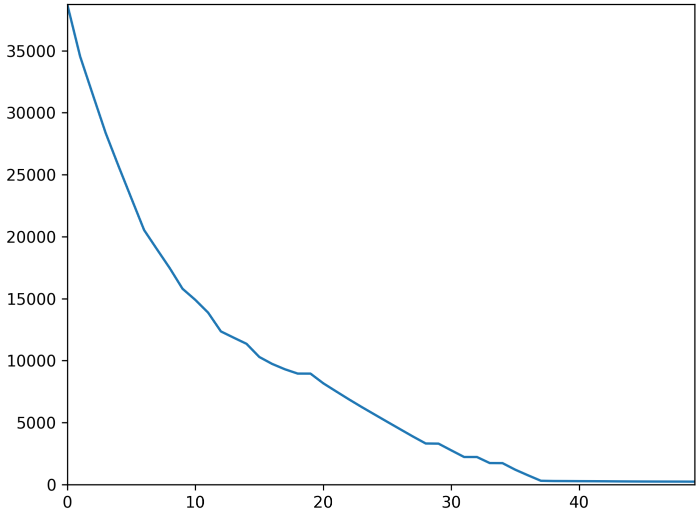


Figure 4.4: The total sum of squared jerks, $\mathbb{J}$ in Eq.(4.2), on the trajectory is effectively reduced during the iterations of our method. The value has been reduced by 99.4% on the final result.

4.3.3 Collision-free Verification

After computing an optimized path $\mathcal{P}$, we need to verify the collision-free at every configurations on the path. This is implemented by applying an advanced collision-detection library (e.g., the FCL library [51]) at every point $\mathbf{q}_i \in \mathcal{P}$.

If collision is found at $\mathbf{q}_d$, we adopt the following projection method to correct it:

- Re-sampling the rotational angle θ around the corresponding waypoint $\mathbf{x}_d$ in a very dense rate (i.e., with the step of $\pi/500$);
- computing the collision-free IK solutions $\{\hat{\mathbf{q}}_d^k\}$ for these dense samples;
- selecting the closest one to serve as a projected solution as:

$$\mathbf{q}_d = \arg\min_{\{\hat{\mathbf{q}}_d^k\}} \|\mathbf{q}_d - \hat{\mathbf{q}}_d^k\|_\infty. \tag{4.13}$$

The infinity norm is employed here to control the maximal variation for all joints. The result of projection may still violate the requirement of maximal jerk. Fortunately, our approximation of Γ is accurate (see next Section) and we did not observe such a scenario in our experiments. In the worse case, we can still split one trajectory into two short trajectories to avoid large jerk in motion.

4.4 Learning-Based Collision Estimation

An efficient method for estimating the collision-indication function $\Gamma(\cdot)$ is needed for solving the jerk-minimization problem by numerical optimization. Collision checking is a computational bottleneck for motion planning. For the general shape of obstacles, there is no trivial mapping from the working space to the configuration space. For the sake of efficiency, we develop a sampling-based approach to learn a function $\tilde{\Gamma}(\cdot)$ to accurately approximate the collision-indication function $\Gamma(\cdot)$. Without loss of generality, collision detection can be formulated as a binary classification problem with $\tilde{\Gamma}(\cdot) < 0$ for collision-free and $\tilde{\Gamma}(\cdot) \geq 0$ for collided configurations. This section first introduces our machine learning method for the representation of $\tilde{\Gamma}(\cdot)$, and then introduces our contact centered sampling strategy used to reduce the required number of training samples.

4.4.1 Approximate Representation of Collision Function

Kernel-based function representations such as *support vector machines* (SVM) and neural networks can be used to generate an algebraic function for $\tilde{\Gamma}(\cdot)$. In our work, we choose SVM as: 1) learning of SVM is a convex optimization problem that can be efficiently solved, and 2) SVMs yields sparser models for high-dimensional non-linear classification problems – i.e., with less number of kernel functions so that the value of $\tilde{\Gamma}(\cdot)$ can be evaluated more efficiently.

Briefly, a SVM algorithm learns a hypothesis function which maps data from an input space to the feature space. Here, the input space is the configuration space $\{\mathbf{q}\}$ and the feature space is the status of collision. Given n sample points with labels obtained by the geometry-based collision detection library, we can learn a *radial basis function* (RBF) based representation of $\tilde{\Gamma}(\cdot)$ as

$$\tilde{\Gamma}(\mathbf{q}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{q}_i, \mathbf{q}) + b \tag{4.14}$$

by using the Gaussian kernel function

$$K(\mathbf{q}_i, \mathbf{q}) = \exp\left(-\gamma \|\mathbf{q} - \mathbf{q}_i\|^2\right).$$

The learning result is the centers of kernels $\{\mathbf{q}_i\}$ as the sub-set of training samples, the coefficients of RBFs $\{\alpha_i\}$ and the value of a bias term b. As an algebraic function is provided here, we can evaluate the gradient of $\tilde{\Gamma}(\mathbf{q}) \approx \Gamma(\mathbf{q})$ by the method presented in [85] when solving the problem defined in Eq.(4.12).

By the property of sparsity in SVM learning, the number of kernels N could be much less than the number of training samples n. It is desirable to obtain fewer kernels so that $\tilde{\Gamma}(\cdot)$ can then be evaluated more efficiently. γ is a parameter specifying the narrowness of the Gaussian, and we use $\gamma = 0.7$ in all our tests. Details of SVM learning and the method for tuning the parameter γ can be found in [86, 87]. Note that, to make the collision-indication provided by $\tilde{\Gamma}(\cdot)$ more conservative, we update the value of the bias term b to $b = b + \epsilon$ after obtaining the solution of SVM learning as $\tilde{\Gamma}(\cdot)$. $\epsilon = 1.0$ is employed in our implementation.

In the applications of robot-assisted manufacturing, the collision-indication function needs to be evaluated and also trained efficiently. The efficient evaluation is demanded as the function is used in the loop of the numerical optimization. The efficient training is also very important as the shape of a workpiece under fabrication (also considered as obstacles) is changed from time to time. By using the routine developed in the work presented in the previous chapter [58], the collision-indication function needs to be updated when the fabrication process moves from one working surface layer to the next one. Similar concept of working surface layers can also be found in the subtractive manufacturing [88]. In general, we need to train a function $\tilde{\Gamma}(\cdot)$ for each working surface when conducting the robot-assisted manufacturing. To reduce the training time, an effective way is to use less number of training samples. We develop a special sampling strategy for this purpose below. Our method can construct the training data-set, which is more capable to identify the boundary between collided and collision-free regions in the configuration space.

4.4.2 Sampling Strategy for Training

Inspired by the active learning method with a coarse-to-fine iterative sampling refinement strategy presented in [80], we first generate sparse samples in the configuration space to capture the large scale topology of the indication function. This function is later refined by adding more selected samples near the decision boundary.

C-space of Contact

To generate more effective training samples, we introduce a concept of contact *configuration space* (C-space) as a set of all configurations where the robotic system's tool touches a working surface $\mathcal{S}$. Given a forward kinematic function $\mathbf{f}(\cdot)$ of the robotic system, the contact C-space of the working surface $\mathcal{S}$ is defined as:

$$\mathcal{Q}_{cont} = \{\mathbf{q} \mid dist(\mathbf{f}(\mathbf{q}), \mathcal{S}) = 0, \forall \mathbf{q} \in \mathbb{R}^L \}$$

with $dist(\cdots)$ being the distance function.

Samples are generated around $\mathcal{Q}_{cont}$ by the steps of initialization and the contact centered refinement as presented below. A projection operator $\Upsilon(\cdot)$ is developed to project a general configuration $\mathbf{q}$ onto the C-space of contact by solving the following minimization problem as

$$\begin{aligned} \Upsilon(\mathbf{q}) &= \arg\min_{\mathbf{q}^*} \|\mathbf{q}^* - \mathbf{q}\|_2^2 \\ s.t. \quad & dist(\mathbf{f}(\mathbf{q}^*), \mathcal{S}) = 0. \end{aligned} \tag{4.15}$$

In our implementation, the solution of $\mathbf{q}^*$ is computed by the sequential quadratic programming and the distance function is efficiently evaluated by the *Proximity Query Package* (PQP) library [56].

Initial Sampling

Sparse samples in $\mathcal{Q}_{cont}$ are initially generated by sampling the working surface $\mathcal{S}$. Specifically, we randomly sample m points on $\mathcal{S}$. At each sample point, we can generate a quaternion by using the surface normal and a random angle θ as mentioned in Eq.(4.3). The corresponding configuration of each quaternion can be obtained by the IK calculation, and the collision status is obtained by the geometry-based collision detection (e.g., the *Flexiable Collision Library* (FCL) presented in [51]). This gives the initial set of training samples – all from the C-space of contact.

Up-scaling and Refinement

Two steps are employed to generate more samples in the nearby region around the the contact C-space $\mathcal{Q}_{cont}$.

- **Up-scaling:** The purpose of this step is to generate nearly uniform samples in the nearby region of $\mathcal{Q}_{cont}$ for capturing the topological structure of the collision-indication function $\Gamma(\cdot)$. Randomly sampling the working surface will not enable this uniformity in the configuration space as the mapping of IK can be very complicated. Directly generating random samples in the configuration space would however require much more samples to capture the structure: most of samples generated in this way would be far away from the C-space of contact, $\mathcal{Q}_{cont}$. Differently, we generate more samples near the C-space of contact by up-sampling the initial set of samples. When the distance between an existing sample and any of its k-nearest neighbors is larger than a threshold τ_α, a new sample is generated in the middle. In effect, 50% of the newly generated samples will be projected onto the C-space of contact, $\mathcal{Q}_{cont}$. The up-scaling is repeated until no new sample can be generated under the density control of τ_α (e.g., $\tau_\alpha = 0.8$ is chosen in our implementation by empirical tests).
- **Refinement:** After up-scaling, a step of boundary-aware refinement is applied to generate samples for learning a more precise decision boundary when approximating the collision-indication function $\Gamma(\cdot)$. Similarly, this is based on searching the k-nearest neighbors of existing

samples. When a sample and its neighbor have different collision labels, we generate a new sample if their distance is larger than a threshold τ_β. A denser sampling is desired along the boundary; therefore, $\tau_\beta < \tau_\alpha$ is used (e.g., $\tau_\beta = 0.05$ in our implementation). Again, the 50% of the newly generated samples will be projected onto $\mathcal{Q}_{cont}$, and the refinement is repeatedly applied until no new sample can be generated under the density control of τ_β.

These two steps are repeatedly applied until the specified total number of samples has been generated. For the nearest neighbors search, $k = 20$ is used in all our examples. The pseudo-code of these two steps can be found in Algorithm 7. As can be found in the following sub-section of analysis, the approximation function generated by SVM can better capture the indication function with the help of much smaller number of training samples.

4.4.3 Analysis

We employ a 3-DOF planar redundant robotic arm as an example to analyze the effectiveness of our sampling strategy in a 2D path tracing setup (see the left of Fig. 4.5).

To follow the 2D path displayed in green color, the three joints of this robot form a C-space of contact in the configuration space (see the right of Fig. 4.5). When presenting obstacles as the red objects, the blue regions denote the collision-free configurations on the C-space of contact. In this analysis, we study how our sampling-and-learning method can effectively capture the boundary between collision-free and collided regions.

Figure 4.6 shows our results comparing to those of SVM-learning by random samples. The ground-truth collision-indication function on the C-space of contact is shown in Fig. 4.6(a). The progressive results of our sample generation algorithm are given in Fig. 4.6(b), from which it is easy to find that samples generated by our method properly capture the boundary of the indication function after the steps of up-scaling and refinement. As can be seen in Fig. 4.6(c), the structure and the boundary of collision-free regions cannot be captured when the same number of samples are generated randomly. In this case, less effective kernels are obtained by SVM-learning. The region of collision-free configurations can be better captured when using more random samples – see the result shown in Fig. 4.6(d), where the same number of effective kernels are obtained by SVM-learning. However, the function learned in this case is still less accurate than ours. In summary, the

Algorithm 7: Generate Samples for Training

```
Input: A working surface $S$.
Output: A resultant set $\mathcal{Q}$ of samples.
/* The step of Initialization */
Randomly generate $m$ samples for $\mathcal{Q}$ with all on $\mathcal{Q}_{cont}$;
repeat
    /* The step of Up-scaling */
    foreach $\mathbf{q} \in \mathcal{Q}$ do
        if $\mathbf{q} \in \mathcal{Q}_{cont}$ then
            Search the $k$-NN of $\mathbf{q}$ as a set $\mathcal{Q}_{NN}$;
            foreach $\mathbf{q}^* \in \mathcal{Q}_{NN}$ do
                if $\|\mathbf{q}^* - \mathbf{q}\| > \tau_\alpha$ then
                    GenerateNewSample($\mathbf{q}$, $\mathbf{q}^*$, $\mathcal{Q}$);
                end
            end
        end
    end
    /* The step of Refinement */
    foreach $\mathbf{q} \in \mathcal{Q}$ do
        if $\mathbf{q} \in \mathcal{Q}_{cont}$ then
            Search the $k$-NN of $\mathbf{q}$ as a set $\mathcal{Q}_{NN}$;
            for every $\mathbf{q}^* \in \mathcal{Q}_{NN}$ do
                if $\|\mathbf{q}^* - \mathbf{q}\| > \tau_\beta$ AND $L(\mathbf{q}^*) \neq L(\mathbf{q})$ then
                    /* $L(\cdot)$ returns the collision
                       status */
                    GenerateNewSample($\mathbf{q}$, $\mathbf{q}^*$, $\mathcal{Q}$);
                end
            end
        end
    end
until no new point can be added into $\mathcal{Q}$;
return $\mathcal{Q}$;
Procedure GenerateNewSample($\mathbf{q}$, $\mathbf{q}^*$, $\mathcal{Q}$)
begin
    $\mathbf{q}_{new} = \frac{1}{2}(\mathbf{q} + \mathbf{q}^*)$;
    /* 50% new samples projected */
    Generate a random $\rho \in [0, 1)$;
    if $\rho \geq 0.5$ then $\mathbf{q}_{new} = \Upsilon(\mathbf{q}_{new})$;
    Add $\mathbf{q}_{new}$ into $\mathcal{Q}$;
end
End Procedure
```

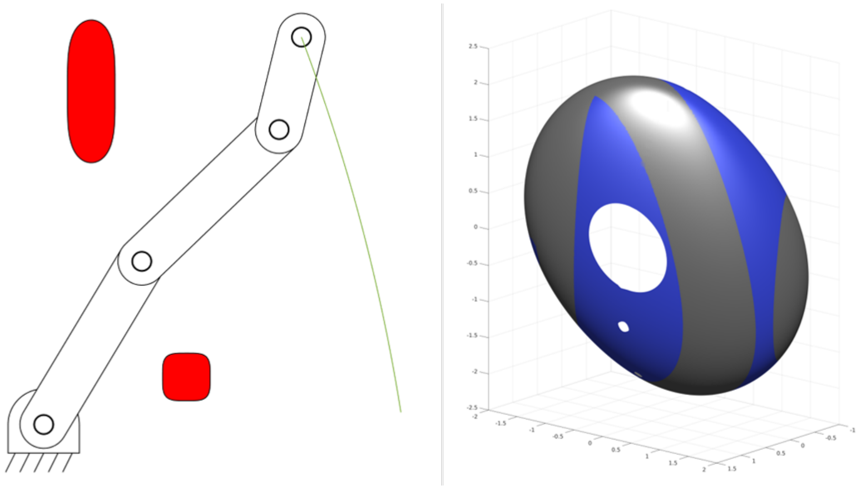


Figure 4.5: A 3-DOF planar robotic arm for tracing a 2D path (green) with obstacles (red). The example is used to study the effectiveness of our sampling strategy for learning the collision-indication function. As shown in the right, the C-space of contact $\mathcal{Q}_{cont}$ is displayed by blue color for the collision-free region (i.e., $\mathcal{Q}_{cont} \cap \mathcal{Q}_{free}$) and gray color for the collided region (i.e., $\mathcal{Q}_{cont} \cap \overline{\mathcal{Q}_{free}}$). Note that, the white regions in the configuration space are not reachable by the robotic arm.

sampling strategy developed in our algorithm can better capture the boundary of an collision-indication function when SVM-learning is adopted. More experiments about the prediction rate and the checking time will be shown in the following section.

4.5 Experimental Results

We implement our algorithm on *Robot Operating System* (ROS) framework with our UR3 based robotic fabrication setup by using C++. All evaluations are executed on a PC with Intel®Core$^{\text{TM}}$i7 processor, 32 GB RAM and GeForce GTX 2070 video card, running Ubuntu 16.04 (Xenial, 64-bits OS). Besides of computational experiments, the performance of our approach has been demonstrated on two different hardware systems for robot-assisted AM (i.e., Fig. 4.7 for a 6-DOF system and Fig. 4.8 for a 8-DOF system).

Our planning algorithm can effectively and efficiently compute a smooth and collision-free trajectory of redundant robot. The quality of fabrication can be significantly improved as the jerk has been optimized on the motion

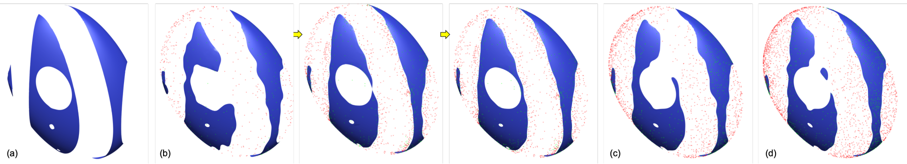


Figure 4.6: Comparisons to demonstrate the effectiveness by using the samples generated by our method to learn a function $\tilde{\Gamma}(\cdot)$ to approximate the collision-indication function $\Gamma(\cdot)$ (a). From the left to right in (b), the progressive sampling results and their corresponding $\tilde{\Gamma}(\cdot)$ are obtained by SVM are shown as: i) after the first round of up-scaling (with $n = 474$ and $N = 348$), ii) after the first round of refinement (with $n = 1527$ and $N = 714$), and iii) the final result of sample generation (with $n = 1698$ and $N = 779$) after a few iterations. The learning results are worse than ours if random sampling is employed to generate (c) the same number of samples as ours and (d) the same number of kernels as ours, where (c) 1698 samples only can result in 517 effective kernels and (d) needs 2553 samples to obtain 779 effective kernels.

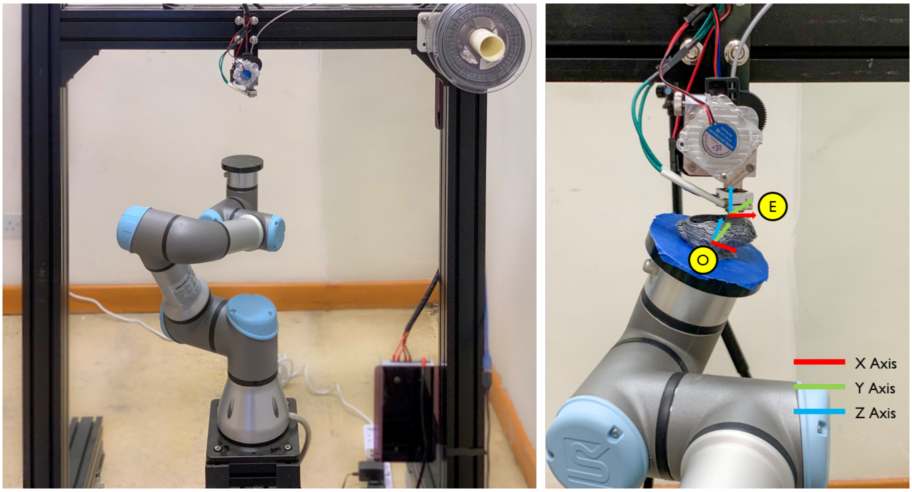


Figure 4.7: A hardware setup of our robot-assisted AM system with 6-DOF. (Left) The UR3 robot-arm based setup with a fixed material-extruder for better material adhesion. (Right) With the help of the relationship between the extruder frame E and the frame of workpiece O, an analytic inverse kinematic solver can be employed to obtain configurations in the joint space from a quaternion determined by rotating the frame E around the orientation given in a waypoint.

trajectories. More details can be found in the reported experimental tests below.

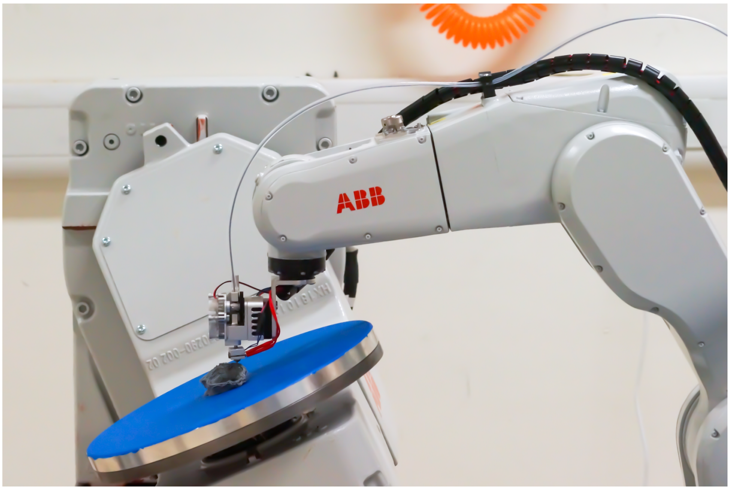


Figure 4.8: A hardware setup of robot-assisted AM system with 8-DOFs consisting of a 6-DOF ABB IRB1200-7/0.7 robotic arm and a 2-DOF IRBP A-250 tilting table.

4.5.1 Learning Results of Collision-Indication Functions

In our implementation, the libSVM library [89] was used for SVM-learning. The effectiveness of our sample generation method for SVM-learning based estimation of collision-indication function has been demonstrated by a planar redundant robot in Section 4.4.3 above. Here we further study its performance in robot-assisted fabrication by using 3D tool-paths. To quantitatively measure the accuracy of $\tilde{\Gamma}(\cdot)$ for approximating $\Gamma(\cdot)$, we evaluate the following *true-negative-ratio* (TNR) metric based on samples of verification.

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}, \tag{4.16}$$

where TN (*True Negative*) denotes the number of samples with $\tilde{\Gamma}(\cdot) < 0$ that are really collision-free while FP (*False Positive*) gives the number of samples with $\tilde{\Gamma}(\cdot) \geq 0$ that however will *not* lead to collision or contact. TNR provides the percentage of samples that are truly collision-free among all samples with negative value returned by the indication function $\tilde{\Gamma}(\cdot)$, the larger the better.

We generate two different types of samples to verify the accuracy of a classifier on different *working surfaces* (WS) as shown in Fig. 4.9. In the first type, verification samples are generated at the nearby regions of the contact-manifold to simulate the situations while computing the numerical optimization. In the second type, verification samples are generated on the contact-manifold by the method of initial samples presented in Section 4.4.2. In both types of tests, $100,000$ verification samples are employed for all examples. The true status of the samples are generated by the geometry-based collision-detection library. The resultant statistics can be found in Table 4.1. It is easy to find that the accuracy of our collision-indication function is much higher than the classifier generated by SVM-learning from random samples. Note that, for conducting a fair comparison we also provide the results of a random-sampling based classifier with the same number of kernels (i.e., the same value of N), which needs much more samples. The nearby sets of verification samples are more similar to the situation that happens during numerical optimization. Comparing to the geometry computation based collision-detection technique (e.g., FCL library [51]), the evaluation of $\Gamma(\cdot)$ with $N = 887$ kernels is $5\times$ faster if only collision-check is needed. However, the gradient of the collision-indication function needs to be evaluated in the numerical optimization (i.e., Eq.(4.12)). To provide a similar function by the geometry-based collision detection, we need to evaluate the distance to obstacles. In this case, our method is around $220\times$ faster.

Table 4.1: Statistics for the Accuracy of Collision-Indication Functions Generated by SVM-Learning

Verification Type	Nearby Region		C-space of Contact	
TNR	WS1 Fig.4.9(a)	WS2 Fig.4.9(b)	WS1 Fig.4.9(a)	WS2 Fig.4.9(b)
Our Method ($n = 3020$ & $N = 887$)	0.963	0.944	0.969	0.956
By Random Samples ($n = 3020$ & $N = 479$)	0.894	0.876	0.942	0.923
By Random Samples ($n = 5842$ & $N = 887$)	0.889	0.883	0.951	0.943

4.5.2 Results of Jerk-optimized Trajectories

In this sub-section, we show the resultant motion trajectories generated by our jerk-optimized planning method. The first example is a tool-path as shown in Fig. 4.9(a) for the 6-DOF robotic system. The progressive results for optimizing the trajectory have been shown in Figs. 4.3 and 4.4. It can be observed that our optimization approach can reduce both the maximal jerk and the total sum of squared jerk by $83.6\% - 95.8\%$ and 99.4% respectively.

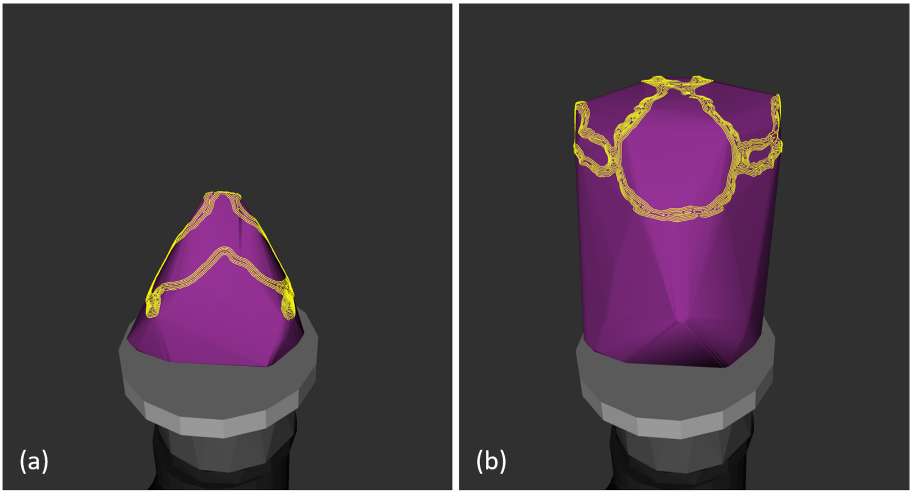


Figure 4.9: Two working surfaces with tool-paths used in our experimental tests: (a) one working surface layer of the armadillo model with 2545 waypoints – its resultant trajectory is shown in Fig. 4.10 and (b) working surface layer of the armadillo model with $4,681$ waypoints having the optimized trajectory given in Fig. 4.11.

Fig. 4.10 shows the path's angular values on all six joints before and after the optimization, where the zoom views clearly show the improvement on smoothness. In the second example, the tool-path as shown in Fig. 4.9(b) is the target to be realized on the UR3-robot based hardware platform for FDM-based AM. The trajectories in joint space before and after optimization are given in Fig. 4.11, where the maximal jerk on all joints have been reduced by up to 95%.

We also compare the results generated by our method to the graph-search based method with a denser sampling – i.e., the nodes in a ladder are generated by every degree for the value of θ in Eq.(4.3). This is actually the method used in [58]. Detail computational statistics of our trajectory planning algorithm in the robot-assisted AM application can be found in Table 4.2. It can be observed that our method generates a trajectory with much lower jerks (both the maximal jerk and the total jerks) while being $10\times$ faster. This demonstrates both the effectiveness and the efficiency of our approach.

4.5.3 Robot-Assisted AM

We also test the trajectories computed by our method in physical experiments using robot-assisted AM. To explicitly show the quality improvement in the

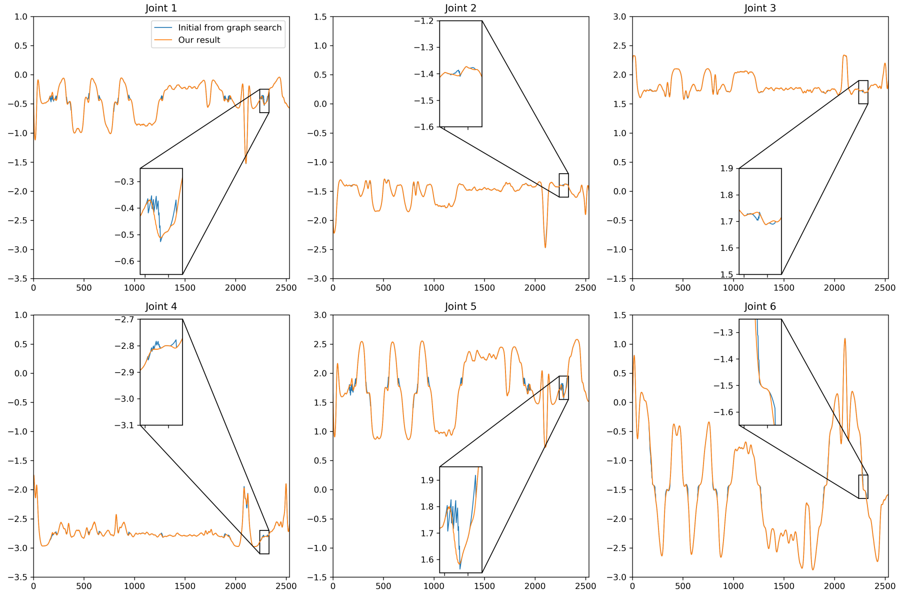


Figure 4.10: The comparison between the initial trajectory from graph-based search and after applying local refinement for each joint on the 6-DOF system. The zoom views shows clearly the smoothness improvement after the local refinement. The tool-path is as shown in Fig. 4.9(a).

real fabrication process, we choose an example tool-path for FDM-based AM on a planar layer as specimen. The specimen are fabricated on the 8-DOF system (Fig. 4.8), where the change of the maximum jerk at each joint during the iteration is shown at the top row of Fig. 4.12. The bottom row of Fig. 4.12 gives the results of FDM-based AM by a graph-search based path (left) and the jerk-optimized path (right). Unwanted blobs can be clearly observed on the path with large jerks, while the jerk-optimized path leads to much smoother material deposition. For the sake of a better illustration, planar tool-paths are conducted in this experiment to demonstrate the influence of large jerk in AM.

The dynamic difference between the trajectories before and after optimization can be more clearly observed in the video (`https://youtu.be/e8ISmh9MPrE`). In summary, the improvement of both the motion smoothness and the quality of fabrication that can be generated by our jerk-optimized trajectory planning algorithm is very significant.

The total time required by our trajectory planning approach is much shorter than the total time of AM process, which is a significant improvement

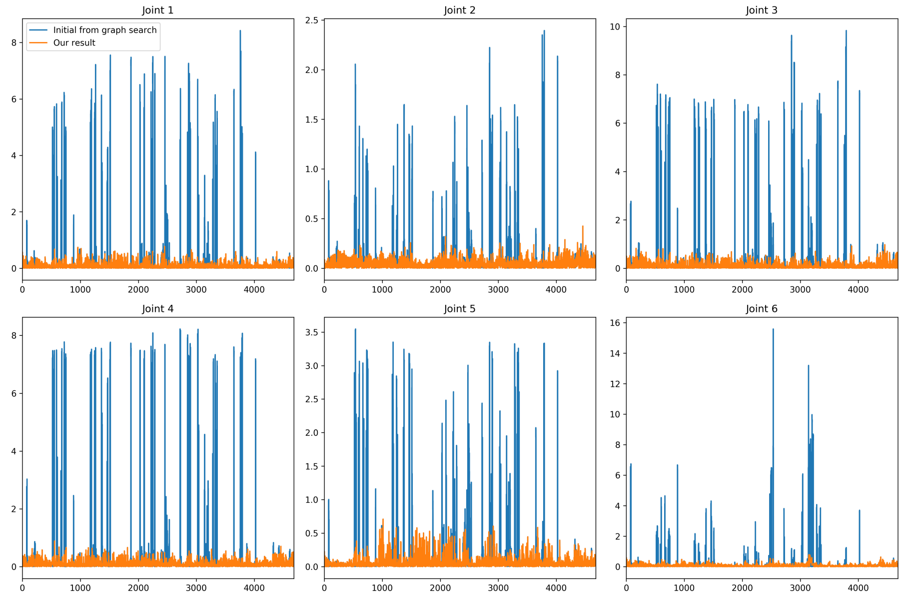


Figure 4.11: The comparison of jerk (the absolute value) between the initial trajectory from graph-based search (displayed in blue color) and after applying local refinement for each joint (displayed in orange color), where significant improvements can be easily observed. Again, this is implemented on the 6-DOF system, and the tool-path is as shown in Fig. 4.9(b).

compared to earlier work presented in previous chapter. On average, a 20× speedup is achieved. For instance, the armadillo model shown in Fig. 4.1 contains more than 300 curved layers and more than 50 layers totalling 2000+ waypoints. The original algorithm presented in [58] needs about 40 hours for motion planning. With the help of the new algorithm presented in this chapter, the total time for planning has been reduced to only 2 hours. For a tool-path with $2,545$ waypoints, the computation can be completed in 116.43 sec., which is much shorter than the fabrication time for the curved layer of this path – i.e., around 460 sec. Motion planning is no longer a bottleneck for robot-assisted AM as we can compute the trajectory of the next layer while working on the current one.

4.5.4 Limitations

Our method is an approach based on local processing so that a more optimal solution can be found by global methods (e.g., the TrajOpt approach [60])).

Table 4.2: Statistics of Computation

Example		**Our Method**				**Dense Graph**	
Path	Wpt. Num.	Computing Time (sec.)			$\mathbb{J}^*$ Eq.(4.2)	Time (sec.)	$\mathbb{J}^*$ Eq.(4.2)
		Init.	Optm.	Total			
WS1	$2,545$	86	30	116	0.232	$1,837$	1.34
WS2	$4,681$	265	58	322	0.697	$3,283$	3.82

	Joint	Maximal Jerk on Path		Resultant Maximal Jerk
		Before Optm.	After Optm.	
WS1	1	15.16	0.76	1.85
	2	5.13	0.80	1.09
	3	5.92	0.97	1.75
	4	6.32	0.97	1.35
	5	21.87	0.94	2.24
	6	17.72	0.74	5.26
WS2	1	8.43	0.77	2.23
	2	2.39	0.42	2.55
	3	9.83	0.97	4.18
	4	8.22	0.89	1.58
	5	3.55	0.71	1.98
	6	15.59	0.78	3.25

*The value of $\mathbb{J}$ is reported at the unit of $\times 10^3$.

When working on a toolpath with 60 waypoints (the first 60 points of WS1), the resultant trajectory with smaller total sum of squared jerks ($\mathbb{J} = 1.64$) can be obtained from this TrajOpt method optimizing the whole path together. The result of our method is $\mathbb{J} = 7.27$. However, the major merit of our approach is its capability to handle a path with large number of waypoints, which is hard to be processed by existing methods. When applying the TrajOpt approach to a toolpath with many waypoints (e.g., the whole WS1), the optimizer is terminated by reaching the penalty iteration limit – it means that the optimization is in fact unsuccessful. The best motion generated by TrajOpt is not well optimized (see Fig.4.13 for a comparison with our approach).

It is also interesting to study the robustness of our approach by adding noises to the orientations of waypoints. Specifically, each orientation can be mapped to a point on the Gaussian sphere, and random noises are added within a range of 5, 10 and 15 degrees respectively in three tests taken on the 8-DOF

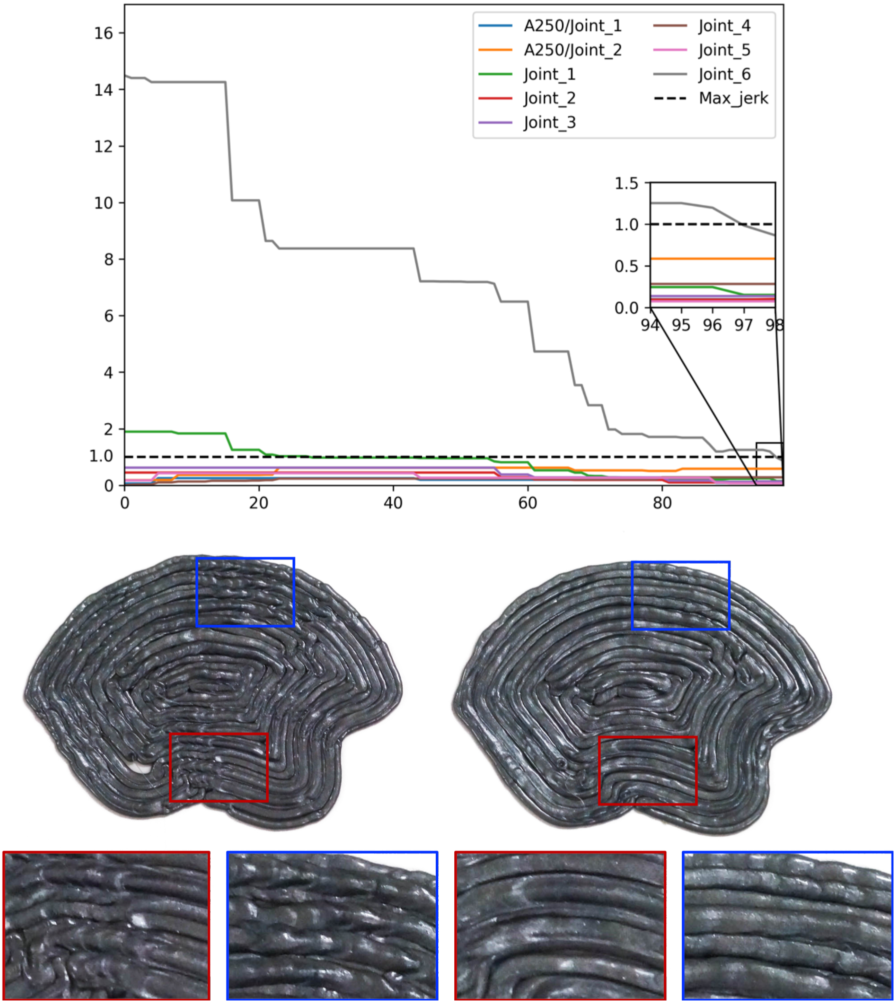


Figure 4.12: The experimental test conducted on a robotic system with 8-DOFs (Fig. 4.8). (Top) The progressive results of jerk-optimization can effectively reduce the maximal jerk on all the eight joints to be less than a threshold 1.0. (Bottom) The results of fabrication by a path with large jerk from graph-search (left) and a jerk-optimized path (right). Unwanted blobs can be observed on the result generated by a path with large jerk as the material deposition is not smooth.

system. The performance of our approach on noisy input is given in Fig. 4.14. The maximum jerk cannot meet the constraint as less than 1.0 although the overall jerk $\mathbb{J}$ can always be reduced significantly. This is considered as the major limitation of our approach.

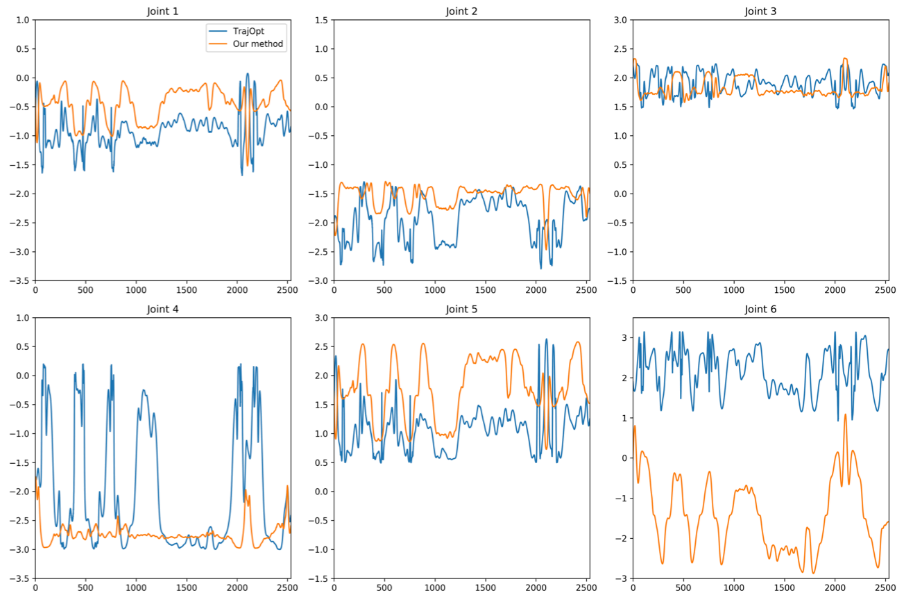


Figure 4.13: When applying the TrajOpt approach [60] to the whole toolpath WS1, their best result is much worse than ours – see the joint angles shown here.

4.6 Conclusion and Discussion

A novel sampling-based framework for planning discrete-time constrained trajectories on redundant robots is presented in this chapter. The major technical contributions include 1) a local filter for jerk minimization while considering other hardware-oriented constraints for feasibility, 2) a greedy algorithm to be applied to a path with many waypoints and 3) an adaptive sampling strategy for effectively learning a collision-indication function with high accuracy. With the help of these techniques, the overall approach can efficiently minimize the total jerk and reduce the maximal jerk. The motion planning solution presented in this chapter is $40\times$ faster than the method in the previous chapter when being applied to all tool-paths of the Armadillo model – the model shown in Fig.4.1.

The performance of jerk-minimized trajectory has been tested in the application of robot-assisted AM using a setup with one robotic arm and one tilting table. The results of the experiment tests are very encouraging, where the fabrication quality in terms of smoothness has been clearly improved while the time efficiency is ensured.

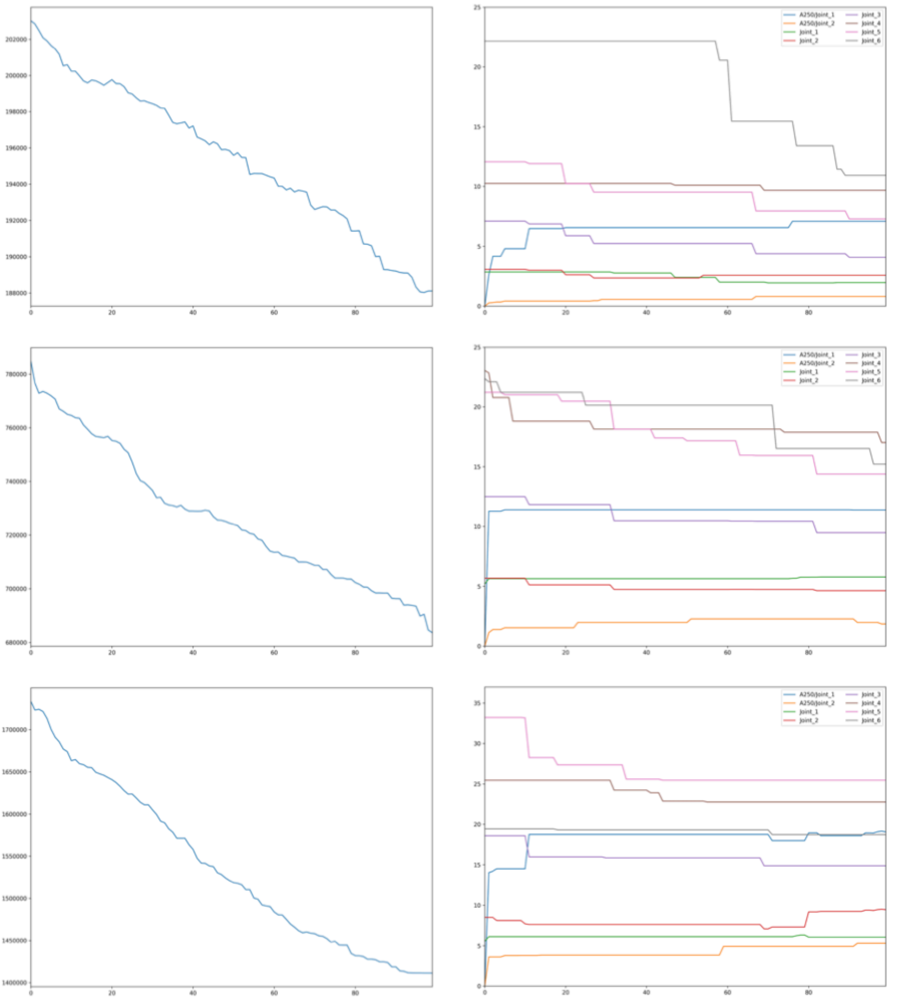


Figure 4.14: Robustness tests conducted on the 8-DOF setup. Left column shows the total sum of squared jerks, $\mathbb{J}$ in Eq.(4.2) and right shows the change of the maximum jerk at each joint during the iterations of our method, from top to bottom are with different noise level within 5, 10 and 15 degrees.

5

Robot-assisted AM: Reinforced with Continuous-Fibers

The future shape of AM technology is explored in this chapter, which is the robot-assisted AM combined with composite material [1]. As discussed in the Chapter 1, the lagging mechanical performance of AM parts limits the freedom of functionality design. A new in-situ fiber reinforcement approach is proposed in this chapter to mechanically enhance AM parts by combining the curved layer based material deposition and continuous carbon-fiber material. Notable enhancement has been observed in the physical experiments, opening the gate to future research towards this new type of AM approach.[5]

5.1 Introduction

Not only the lagging mechanical performance of AM limits the designers' freedom, but it also constrains its usage towards industrial applications. Numerous attempts have been made to enhance the mechanical performance of AM parts by means of making new materials [90], optimizing structures [91] or changing the environment of fabrication [92]. On the other aspects, effort has been made by [93] to add epoxy (one kind of thermoset) to fill up the voids of thermoplastic part. Another method proposed by [94] is to add short fibers in the feedstock to enhance the overall mechanical properties of epoxy-based polymer. Similarly, 3D printed powders are used by [95]. [96] studied how to add short fibers into thermoplastics. Specifically, good

---

[5] This chapter has been published as: Chi-Chung Li, **Chengkai Dai**, Wei-Hsin Liao, and Charlie C.L. Wang, "Towards Direct Deposition of Continuous-Fibers on Curved Surfaces", *Recent Advances in Additive Manufacturing*, Chapter 4, 2020. Note: a few small corrections and/or clarifications have been made to the original published text.

alignments of short fibers can be achieved in fabricated thermoplastics to have significantly enhanced tensile strength and modulus. *Carbon Fiber Reinforced Polymer* (CFRP) composite filaments were prepared from carbon fiber and ABS in extrusion processes by [97]. The study of [98] has demonstrated a method that can align the fibers in the desired 3D architecture by using ultrasonic forces.

Building upon the results of using short fibers in AM, it was hypothesized that further enhancement in performance is possible by depositing continuous fiber instead of chopped short fibers. A design oriented method has been investigated by [99] to manually reinforce thermoplastic parts by carbon fibers and epoxy at the regions with high stresses in AM process. In another manually reinforced case study taken on *Polylactic Acid* (PLA) molded plastic and printing filaments by [100], significant improvement in both strength and stiffness can be observed. Shortly speaking, reinforcement with continuous-fiber is a very attractive means for further enhancing the mechanical properties of thermoplastics fabricated by AM.

Besides of reinforcement provided by continuous-fiber, more and more research has been conducted to align the filament according to the external loading to be added. Specifically, it has also been argued that making a structure (or tool-paths) conformal to the axial surface or the curves of principal stresses can help to fabricate a model stronger in mechanical properties [101, 102, 103].

The purpose of the work in this chapter is to explore the possible methodology to realize the direct deposition of continuous-fibers in a sandwich structure on curved 3D surfaces. Preliminary tests have been conducted to demonstrate the performance improvement after

1. converting a planar-layer-based filament deposition into a curved-layer-based fabrication, and
2. further reinforcing continuous carbon fibers between the curved layers of plastic filaments.

Physical experiments are conducted on a hardware setup with 6 *degrees-of-freedom* (DOF) motion provided by a robotic arm. With the help of such a hardware platform, we are able to reinforce fabricated parts by a process of continuous-fiber deposition between layers of PLA matrix in AM.

5.2 Related Work

As we already reviewed the relevant research of high DOF and robot-assisted AM in the previous chapters. This section mainly focus on the literature of the progress of using continuous-fiber in AM.

For the research of continuous-fiber reinforcement, [104] categorized different production strategies of continuous-fibers into three types: I) pre-impregnation, II) in-nozzle impregnation and III) direct deposition. We review relevant work below by following these categories.

In aerospace industry, Automated Fiber Placement (AFP) has been developed to fabricate fiber composites. Continuously pre-impregnated (pre-preg) fibers can be heated and pressed against a model (or existing layers of the laminate) in thermoplastic AFP processes [105]. The matrix materials around fibers are fused together during the fabrication process. Required composite with high performance is usually manufactured tape-by-tape and then layer-by-layer. As a small-scale counterpart, Mark Two 3D printer invented by [106] is able to deposit filaments reinforced with continuous-fiber in FDM. A special extruder is developed in their system to extrude filaments with pre-preg carbon fibers and produce 3D models in the conventional line-by-line process of layered manufacturing. Currently, the cost per unit volume of pre-preg carbon fiber [107] is USD$\$3/\mathrm{cm}^3$. It is about 10 to 20 times higher than that of typical carbon fibers [108]. The fact that pre-preg carbon fibers are proprietary [109] and relatively expensive blocks the wide usage of this technique.

For the second type of continuous-fiber embedding, recently attempts have been made to achieve it in FDM by inserting a narrow strand of fiber into the back opening of a specially designed nozzle while the filament is extruded along with the molten thermoplastic. When cooling down, fibers are embedded in the strand of thermoplastic and thereby reinforces a fabricated part. The whole process is an integration of impregnation and AM process; therefore, the approach is referred as in-nozzle impregnation. [110] recently developed a method that impregnates fibers with filament within the heated nozzle of 3D printers, where thermoplastic filaments and continuous fibers are supplied separately to the nozzle. [111] conducts a similar strategy to reinforce fibers into fabricated models and the maximum flexural strength of 335MPa and the flexural modulus of 30GPa can be observed on their fabricated composite specimens with 27% fiber content. In the experiments of [112], further enhanced mechanical property is found by using the preprocessed carbon fiber

Table 5.1: Comparison of different production strategies for continuous-fiber

Strategy	Pre-impregnation	In-nozzle Impregnation	Direct Deposition
Width of fiber	*Narrow (Strand)*	*Narrow (Strand)*	*Wide (Strand/Tape)*
Material Cost	*Higher*	*Lower*	*Lower*
Availability	*Proprietary*	*Non-proprietary*	*Non-proprietary*
Depos. Speed	*Slower*	*Slower*	*Faster*

with polylactic acid sizing agent. All these approaches using very thin carbon fiber bundles (e.g., the one made up of 1000 single carbon fibers in [112]).

In the above two types of continuous-fiber reinforcement, the width of fibers to be used is constrained by the mechanical design of nozzle in extrusion head. As a result, the strand based deposition could be very slow. If carbon fiber could be directly deposited on the fabricated part, the fiber deposition speed could be much higher (see the comparison listed in Table 5.1 – Comparison of different approaches for reinforcing continuous-fibers in AM). In the rest of this chapter, we will introduce our preliminary research towards this direction of direct carbon fiber deposition.

Along the other thread of research, it has been observed that the mechanical strength of parts made by FDM is strongly affected by the arrangement of thermoplastic strands. The adhesive strength across plastic strands is much weaker than the longitudinal strength along the strands. It has been advised to keep plastic strands as continuous as possible during the fabrication and orient filaments along the loading direction to maximize mechanical performance. In planar layered AM, choosing the right material accumulation direction can let strands be continuous for simple objects but it does not apply to parts with freeform curved surfaces. Chakraborty et al. [36] proposed curved layer FDM (CLFDM) algorithms to generate non-planar paths for fabricating curved objects; however, only simulation is given in their results. Singamneni et al. [113] applied the curved layer FDM algorithms to fabricate physical specimens to verify this conclusion. Recently, Huang and Singamneni [114] further considered the factor of adaptive slice thickness into the fabrication of curved layers. Testing specimens have been made using CLFDM, and significant enhancement of mechanical performance was observed on these specimens. Besides, the surface finish of a model made by CLFDM can be improved due to absence of stair-case effect. Differently, we consider how to incorporate the material properties of continuous-fiber into the

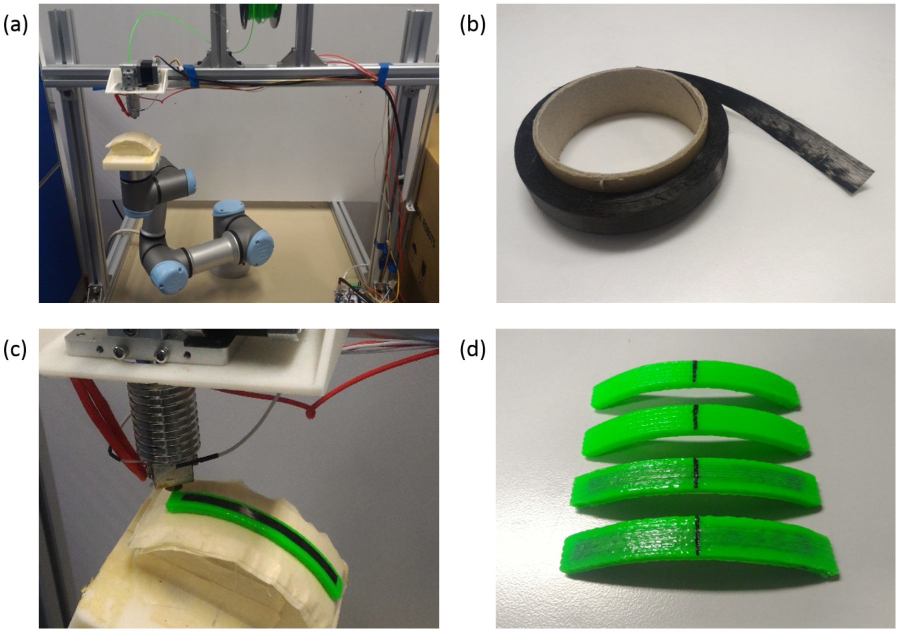


Figure 5.1: The hardware setup in our research for continuous-fiber reinforced AM with 6-DOF motion: (a) 6-DOF platform used for direct deposition, (b) tapes of carbon fibers reinforcements used for AM, (c) accumulating PLA on top of ironed fibers, and (d) samples fabricated by curved deposition with and without continuous-fiber reinforcement.

curved layers of PLA with the help of a robot-assisted system.

5.3 Direct Deposition: A study

In this section, we introduce a direct deposition method to place tapes of continuous carbon fibers between matrix of thermoplastics (e.g., PLA) so that the mechanical strength of a printed model can be reinforced. We wish more work will be motivated towards the direct deposition of continuous-fibers in AM process. Figure 5.1(a) shows our hardware setup with 6-DOF motion used for the direct deposition of continuous-fiber. Note that when planar layers are utilized for the continuous-fiber reinforcement, the direct deposition can also be conducted on the planar motion platform of conventional FDM 3D printer (e.g., the one shown in Fig.5.3(a)).

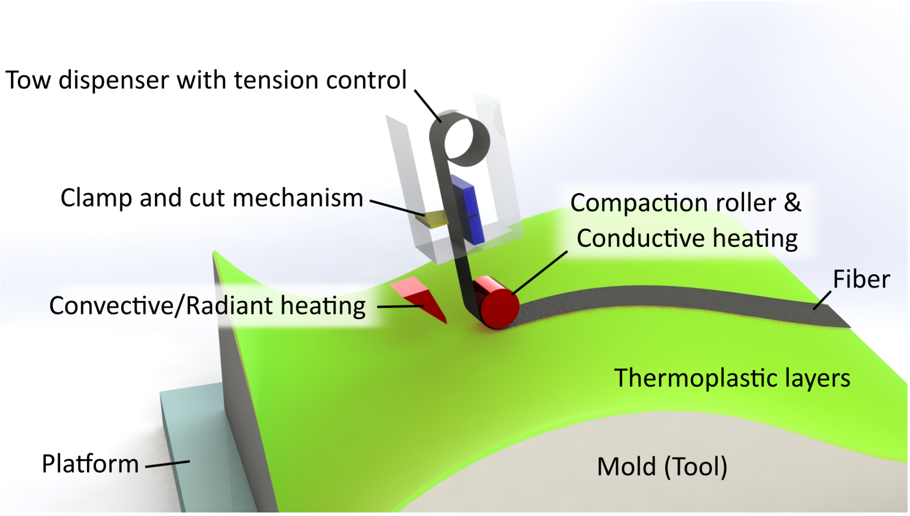


Figure 5.2: Illustration of heated deposition mechanism for continuous-fiber reinforcement in AM process.

5.3.1 Processing Method

Our approach can be considered an a variant of AFP process in AM. Continuous tapes of carbon fibers are placed between layers of thermoplastic matrix – i.e., continuous-fibers and PLA used in our experiments are fused in a sandwich way. A layer of thermoplastic must be deposited as the substrate before the direct deposition of fibers. When a layer of thermoplastic matrix is ready, head of fibers (in tape) is first placed on top of the fused thermoplastics at the desired location and with an orientation normal to the surface. Heat is then applied on the interface between fibers and thermoplastics in a conductive, convective or radiant manner. When the heated region is cooled down (preferably under pressure), the placed fiber will be locally bonded with the thermoplastic. The deposition mechanism, which applies heat and pressure, travels along the length of a tape to continuously bond fibers to the previous layer of thermoplastic substrate.

With the help of 6-DOF motion platform, the head of deposition would trace a planned path on the working surface and keep applying perpendicular pressure to the deposited fibers. At the end of a path, chopper in the mechanism is applied to cut the continuous-fiber into segments (see Figure 5.2 for an illustration of the mechanism). This step of direct deposition is similar to the AFP technique used in the aerospace industry. Differently, AFP only uses pre-preg tow whereas in this approach raw spread tow (that only contains the reinforcing fibers) are directly bonded onto thermoplastic substrate.

To complete the process of direct deposition, a layer of thermoplastic is fused on top of the placed fibers in a way similar to conventional FDM but again with 6-DOF motion (see Figure 5.1(c)). It is noted that height or extrusion compensation has to be considered for the volume taken by the additional fibers. In our practice, the thickness of thermoplastic layer (in the range of 0.3mm to 0.6mm) is much larger than that of fibers (about 0.08mm). No compensation is taken for the fiber layer, and the height increase between thermoplastic layers is conducted by using the uniform offsetting method [115]. After the upper layer of thermoplastic matrix is fused, continuous tapes of fibers have been securely embedded into the fabricated part (see Figure 5.1(d) for the results of fabrication).

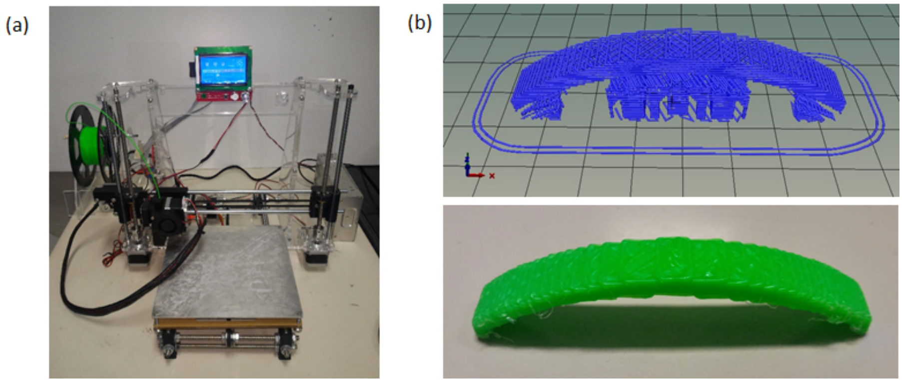


Figure 5.3: Comparison to the specimen fabricated by a conventional AM system with planar motion and (b) the model with curved surfaces fabricated from planar tool-paths.

5.3.2 Detail of Experiments

Materials employed in our experimental fabrication are PLA supplied by Dazzlelight and a spread tow of carbon fiber supplied by Easy Composite Ltd. PLA thermoplastic is chosen for its popularity and ease of fabrication. The spread tow made from Gradfil TR50S 15K carbon fiber tow which contains 6% Polyamide by weight is utilized. In particular, spread tow, an organized and thin bundle of fiber (0.08mm thick), is used as its thickness facilitates the infiltration of plastic and heat conduction in the direct deposition process. The 15mm-wide spread tow is first split into three 5mm-wide ones before

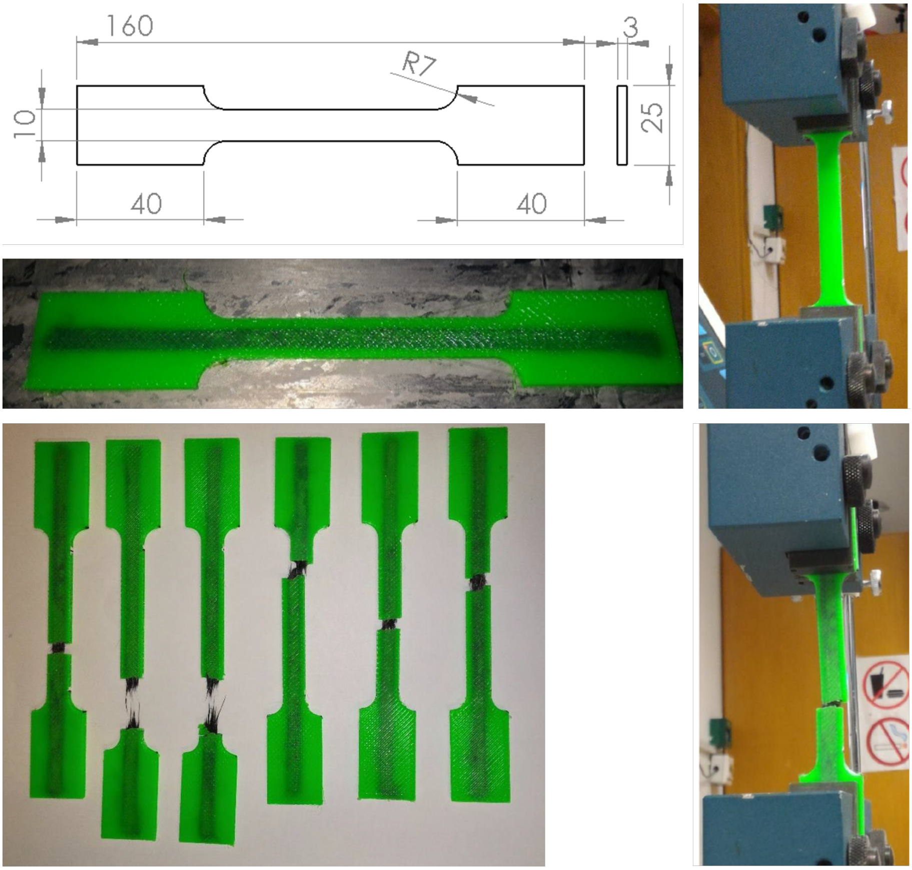


Figure 5.4: Tensile tests taken on reinforced models with planar layers.

application.

In our hardware setup for fabrication, a Universal Robot UR3 robot arm is employed to move the building platform so that 6-DOF relative motion can be realized between the platform (and also the working envelope) and a stationary nozzle. When fabricating models with curved layers, a mold is added onto the building platform to provide a curved surface where the extruded plastic could adhere to. In our tests, the mold is a 3D printed rectangular block with cylindrical surface on the top (80° arc at 60mm radius), and it is covered with masking tape. PLA matrix is printed at 210°. The diameter of nozzle is 1mm and the thickness of PLA layer is controlled at 0.6mm. Note that, our expendable setup actually allows multiple nozzles to be installed on the frame and multi-material fabricating can be realized. For the comparison taken with conventional AM, Reprap Prusa i3 was used as a FDM platform to fabricate testing samples. The machine accumulates PLA with a hotbed at 60°. The temperature of nozzle is set as 230° for planar samples for tensile and bending tests. To be consistent with the samples fabricated by 6-DOF motion platform, 210° was used for PU samples in curved shape (i.e., the one shown in Fig.5.3(b)).

5.4 Results

In this section, we present the results of experimental tests for comparing the mechanical properties of 3D printed specimen in different aspects. Planar and curved models are fabricated to test their mechanical properties with and without reinforcement.

5.4.1 Mechanical Properties of Continuous-fiber Reinforcement by Planar Layers

The first type of tests are taken on models printed by reinforced and non-reinforced planar layers, which are denoted by PR (i.e., planar with carbon-fiber reinforcement) and PU (i.e., planar unreinforced specimen) in the charts of testing results. Both the tensile and the bending tests are conducted.

Twelve dumbbell-shape samples are fabricated and 6 of them are reinforced with 5 layers of 150mm $\times$ 5mm carbon fiber spread tow (as shown in Figure 5.4). Tensile tests for these specimen are conducted on a H5KS Benchtop Materials Testing Machine made by Tinius Olsen. All samples are applied with a preload of 15N and stretched in the speed of 10mm/min. Average ultimate tensile strengths of 1.700kN and 3.535kN are observed on PU and PR samples respectively. The corresponding stress-strain curves generated in these tests are shown in Figure 5.6(a). The estimated Young's modulus are $E_{PU} = $ 1.25GPa and $E_{PR} = $ 2.32GPa – i.e., about 85.6% improvement has been observed on reinforced samples for its stiffness.

For bending tests, 8 rectangular bars with dimension 250mm $\times$ 10mm $\times$ 5mm (thickness) are fabricated with planar layers. Four of the samples are reinforced with 9 layers of 240mm $\times$ 5mm carbon fiber spread tow (as shown in Figure 5.5). All samples are subjected to three-point flexural tests with a maximal loading of 7.27N applied at the middle. By using the simple beam theory, flexural rigidity (EI) of a beam can be obtained by

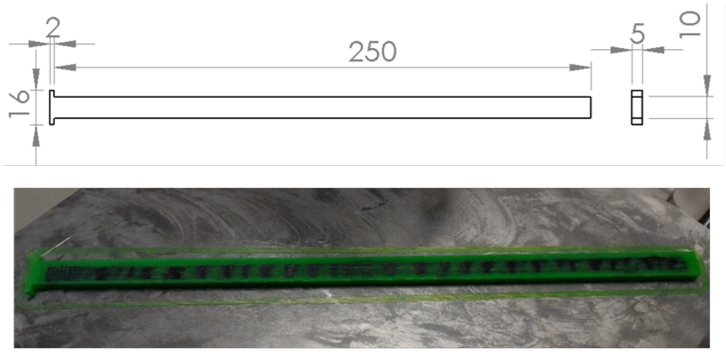


Figure 5.5: Planar specimen (with carbon-fiber reinforced) that are used in the bending tests.

$EI = P \cdot L^3/(48d)$, where P denotes the force applied, L is the distance between the supports and d is measured central deflection. According to the measurements shown in Figure 5.6(b), the flexural rigidity of unreinforced and reinforced bars are $0.351\mathrm{N} \cdot \mathrm{m}^2$ and $0.671\mathrm{N} \cdot \mathrm{m}^2$ respectively. On average, about 91.1% of enhancement in flexural rigidity is achieved on the reinforced specimen.

5.4.2 Mechanical Properties of Continuous-fiber Reinforcement by Curved Layers

After studying the mechanical reinforcement on models fabricated by planar layers, it is more interesting to study the bending behavior on specimen made by curved layers. The tests are taken on curved bars which have an arc in $75°$ with $60\mathrm{mm}$ radius. Width of the bars is set as $10\mathrm{mm}$. 6-DOF motion platform is employed to fabricate 12 such models that each has seven PLA curved layers in the thickness of $0.6\mathrm{mm}$. Six out of the 12 specimen are reinforced by direct deposition with 6 layers of $68\mathrm{mm} \times 5\mathrm{mm}$ spread tow of carbon fibers. Moreover, to compare with the models fabricated by conventional AM, we also made 6 models in the same shape by planar layers as shown in Figure 5.3. Specifically, three different types of fabrication are conducted as explained in Figure 5.7(b).

During the bending tests, compression force is applied on samples with a sharp-edge contact (see the black marks in Figure 5.1(d). Handpi 1000N force gauge is used to measure the force applied and an attached digital caliper is used to measure displacement in the tests. A preload of 10N is applied before

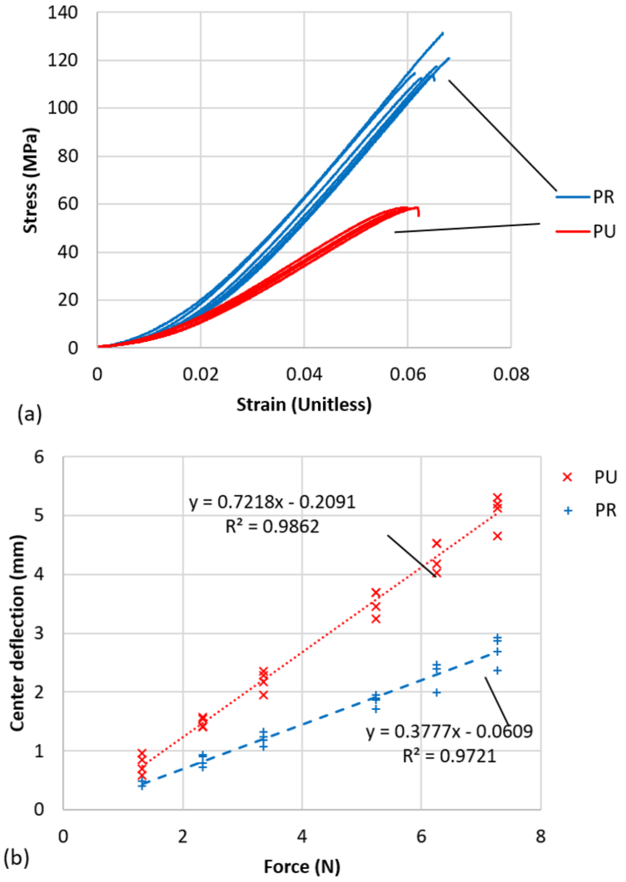


Figure 5.6: Curves of mechanical property generated by physical tests on reinforced planar specimen (PR) vs. unreinforced planar specimen (PU). (a) Stress-strain curves of tensile tests on samples as shown in Fig.5.4. (b) Deflection-force curves of three-points bending tests generated on planar rectangular sample bars as shown in Fig.5.5.

setting both the force gauge and caliper reading to be zero. The preload is used to ensure close and stable contact between the sharp edge and the sample.

Our experimental tests also help to generate the displacement-force diagram of bending as shown in Figure 5.8. A straight line is used to estimate the displacement-force relationship of each type of specimen using least-square regression. Here the reciprocal of slope of fitting result, in terms of required force to create unit displacement, is conducted as a measurement of bending resistance. The observed average reciprocals of

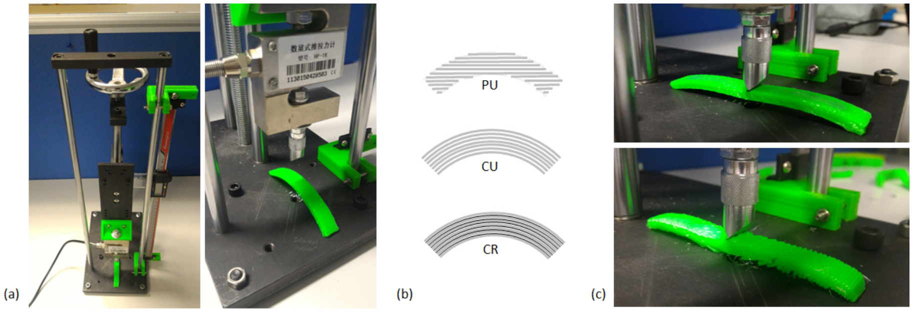


Figure 5.7: (a) The hardware setup of bending test applied to the specimen. (b) Specimen in curved shape fabricated by different strategies: PU – by planar unreinforced layers, CU – by curve unreinforced layers, and CR – by curve reinforced layers. (c) Under the same loading, delamination can be found on the model fabricated from planar layers (bottom).

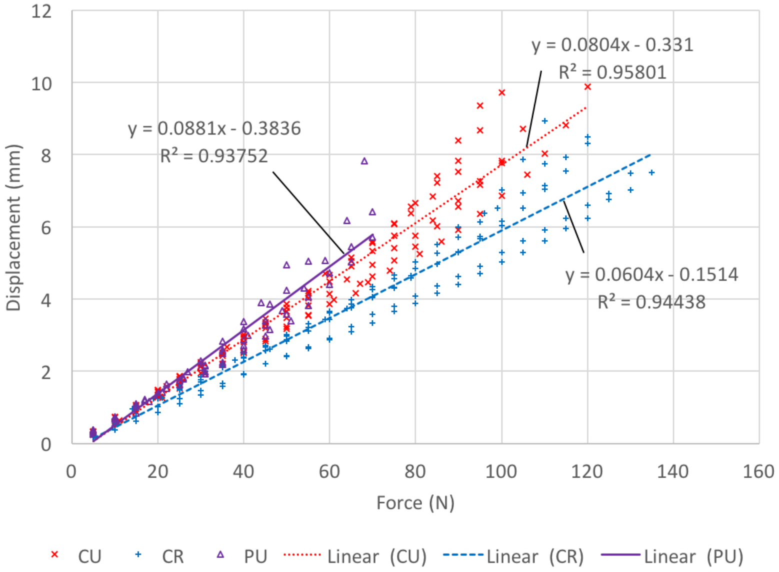


Figure 5.8: Displacement-force diagram generated by bending tests on PU, CU and CR specimen.

PU, CU and CR specimen are $11.35\mathrm{N}/\mathrm{mm}$, $12.43\mathrm{N}/\mathrm{mm}$ and $16.54\mathrm{N}/\mathrm{mm}$ respectively. The reinforced samples are 33.06% more resistance to bending over the unreinforced ones. The bending stiffness has been significantly improved by adding continuous-fiber reinforcement. On the other aspect, our experimental results also prove that with an appropriate orientation of filament the mechanical property of a model fabricated by AM can be enhanced – i.e., on both specimen without fiber reinforcement, the ones fabricated with curved layers (CU samples) shows 9.50% more resistant to deformation than PU samples.

5.4.3 Fracture Tests under Large Deformation

Fracture in the form of delamination between layers is a phenomenon that can always be observed on the curved models fabricated by planar-layer-based AM. For example, such delamination can be observed PU samples when around 80N loadings are applied – see the bottom of Figure 5.7(c). For the samples of CU and CR, fracture observed only when much larger loadings are applied – i.e., 170N and 198N for CU and CR specimen respectively.

Unlike PU specimen, fracture cannot be observed on CU and CR samples even after bending them into a flat shape. It only occurs after further deforming them into an inverse arc shape. This nice mechanical property benefits from the fabrication of layers conformal to the neutral surface of a beam, which also shows the effectiveness of continuous-fiber reinforcement with the help of robot-assisted AM system.

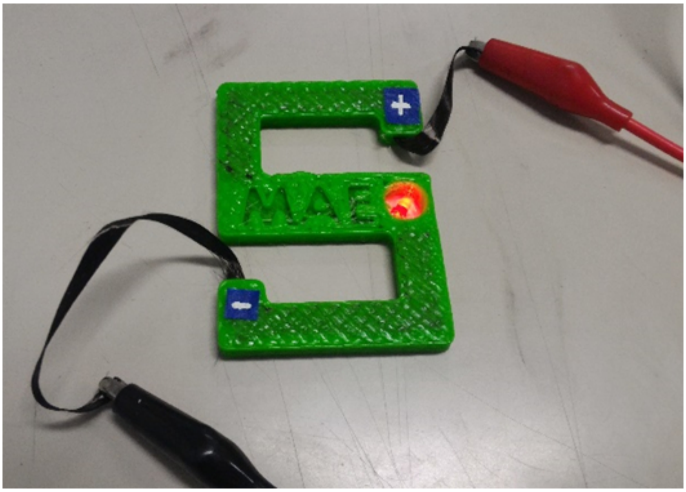


Figure 5.9: Reinforced hook with an LED circuit – continuous carbon-fibers are served as wires to form an electronic circuit.

5.5 Electrical Conductivity

As carbon fiber is electrically conductive, it could potentially be used as wires and form part of an electronic circuit. We have a successful attempt to fabricate a simple LED circuit in a reinforced hook using the direct deposition method proposed in this chapter (see Figure 5.9). This suggested that the embedded fiber could have functions other than structural reinforcements and closer inspection should be given on embedding functional elements in FDM plastics. More discussion about 3D printing electronics can be found in [116].

5.6 Conclusion and Discussion

In all our experimental tests, we fuse tapes of carbon fibers onto the surface layer of PLA substrate manually. Heat is delivered on top of the fiber in a conductive way by soldering iron with series K blade tip [117]. The blade shape in iron's tip helps to deliver heat and pressure evenly across the width of the fiber, which is analogous to the household iron for clothes. Taking references from AFP (such as [105] and [118]), the temperature controlled at the interface and the pressure applied throughout the heating and cooling process are major factors to determine the quality of bonding. If they are further optimized, the mechanical properties of reinforcement achieved by direct deposition of continuous-fibers could be even stronger than what we reported here.

In summary, the new trend of AM is explored in this chapter for fabricating objects with continuous-fiber reinforced by a robot-assisted AM system, which can provide 6-DOF motion to generate curved layers of matrix made by polylactic acid. A preliminary system for direct deposition of continuous-fibers have been realized although there are some steps conducted manually. Notable structure enhancement has been observed in the experimental results of curved layer fabrication with and without fiber reinforcement.

6

Conclusion

In the previous chapters, the results of the four research cycles of this research have been presented. In this chapter, the main contributions of this research are summarized by answering the research questions. In addition, the unsolved problems and possible future research are discussed. This chapter finishes with a reflection on the process of developing the robot-assisted AM system, and the implications of this research.

6.1 Contributions

The research presented in this thesis aims to overcome the limitations of current AM technology by adding rotational motion. In this context, the specific objective of this study is to develop a suitable workflow for the robot-assisted AM system with orientation change. This follows the same workflow as conventional AM, which consists of three steps: slicing, tool-path generation and hardware realization. The research objective was therefore divided into three research questions, which have been answered through four research cycles.

Considering the above-mentioned objective, the main contributions of this thesis can be discussed under two topics, as presented next.

6.1.1 Non-planar Slicer for AM

The slicer is known as the middleman between the 3D model and the AM hardware. The slicer software slices the model into a series of layers and produces execution commands which contain instructions tailored to the hardware. Obviously, answering RQ1 and RQ2 contributes to a new type of slicer software that is suitable for robot-assisted AM system.

RQ1 *How to slice a 3D model in the robot-assisted AM system for overcoming the limitations of AM?*

Slicing the 3D model in the robot-assisted AM system is no longer constrained in planar domain. Although the flexibility of material deposition increases significantly, the change from planar to arbitrarily curved layers tremendously increases the complexity of computations. The additional freedom makes it challenging even to define what the geometry of the layers should be, not to mention the consideration of manufacturing needs and hardware constraints. The answer – to generate optimized slicing results – is proposed by a novel methodology that involves optimizing a scalar field within the volume of the 3D model that represents the fabrication sequence. As illustrated in Chapter 3, the field is constrained such that its iso-values represent curved layers that are supported from below, and a convex surface affording algorithm for collision-free navigation of the extrusion head is presented to meet the hardware constraints. Although this scalar field is dedicated to eliminating the use of support structures, it can also be optimized for surface covering to improve the staircase effect. This field-based approach can also be used to optimize stress distribution for enhancing the mechanical performance, based on the observations made in Chapter 5.

RQ2 *How to generate tool-paths in the 3D domain that can be used in the robot-assisted AM system?*

In this study, the position-continuity and orientation-continuity are identified as the essential factors for tool-paths in robot-assisted AM. In the FDM process, it is difficult to stop and start the flow of material because the material is viscous. A tool-path discontinuity or contour plurality not only forces an on-off switching of the extrusion head, thereby leading the discontinuity of material accumulation, but it also impacts the motion smoothness of the robot. The solution adopted in this study to ensure the position-continuity on curved surface layers is to use a special space-filling pattern called Connected Fermat Spirals (CFS). Previous CFS work focuses on the tool-paths generation in the planar layer. The distance metric in the planar domain can be simply represented by Euclidean distance, which makes it easy to construct the evenly spaced, non-overlapping trajectories. The computation in the freeform surface domain relies on the geodesic metric, which is far more complicated. One contribution of this study is extending this unique pattern to the 3D domain by building an exact geodesic boundary distance via Fast-Wavefront-Propagation (FWP) based on the Mitchell-Mount-Papadimitriou (MMP) method. Furthermore, tool-paths in the freeform surface domain need orientation information to avoid local gouging and to improve fabrication quality. The orientation-continuity is achieved by robustly estimating the normal vector on the sliced surface corresponding to the neighboring tool-path waypoints.

Slicer software should be able to convert the generated tool-path to the G-Code file which contains the extrusion information. In the robot-assisted AM system, both the position and the orientation of the nozzle needs to be specified in the Euclidean coordinate system. Another contribution of this study is that it proposes an extended G-code file format for defining the tool-paths that can be realized the robot-assisted AM system. A detailed description of the extended G-code can be found in the Appendix.

6.1.2 Trajectory Optimization for Robots

Optimizing the robot movement under given tasks is a general problem in the CAD/CAM softwares. Answering RQ3 not only contributes to providing an optimized dynamic behavior for the AM system, but it also can be generally applied in the CAD/CAM softwares for performing other robot-assisted manufacturing applications.

RQ3 *How to plan the robotic motion in the robot-assisted AM system for better realizing the process?*

The major difference in the motion planning problem between the conventional AM system and robot-assisted AM system is that the kinematic redundancy exists in the robot-assisted AM system. This kinematic redundancy makes the motion planning problem in robot-assisted AM non-trivial, as the tool-paths can be realized by a possibly infinite set of the joint space trajectories of the robotic arm. On the other hand, the presence of additional degrees of freedom creates the possibility of optimizing the robot's dynamical behavior. Based on the observation that the jerk induces resonant vibrations in the robot's structure, resulting in unstable material deposition and damaging the fabrication quality, minimizing the jerk of robotic motion is identified as the objective for the motion planning in robot-assisted AM. An answer to RQ3 is the approach proposed in Chapter 4, which can efficiently and effectively plan the jerk-optimized robotic motion in a redundant system. The contribution of this approach is threefold: The first contribution is a local filter for jerk minimization that considers other hardware-oriented constraints on feasibility; the second is a greedy algorithm that is to be applied to a path with many waypoints; and the third is an adaptive sampling strategy for effectively learning a collision-indication function with high accuracy. The generality of this approach makes it suitable for extending to other robotic manufacturing applications where the robots have more DOF than the given tasks demand.

6.2 Unsolved Problems and Future Works

The research presented in this thesis is certainly not the final answer in the field of robot-assisted AM. On the contrary, the limitations have become even more apparent, and much more in-depth research is yet to be carried out in the future. Three major problems have been identified here which are likely to form the basis for future research and development.

6.2.1 System Setup

The system setup that is presented in this thesis is a very basic implementation of the robot-assisted AM system. It is a prototype and still has a long way to go before commercialization, on the other hand, this could be an area where significant improvements could be made to bring FDM to a more mature quality level.

The first thing to improve is the calibration. Calibration in the robot system is a necessary process used to improve the accuracy of the robot. There are several different levels of calibration in robot-assisted AM system: (1) Tool-Center-Point (TCP) calibration, in the context of robot-assisted AM system, is to detect the relative position between the center of build platform or extrusion nozzle and robot's end link frame (or robot's base frame, according to different hardware setup). This relative position directly determines the robot's end pose (as illustrated in Eq.2.3), thereby influencing the accuracy of material deposition. (2) Kinematic calibration, which concerns the geometry of the robot, including link lengths and angle offsets (also known as D-H parameters). D-H parameters are the keys to determine the robot's configuration based on the robot's end pose. Inaccurate robot's configuration will also lead to the inaccuracy of material deposition. In the current implementation, a preliminary touching point calibration method is used [119]. However, the calibration result is not satisfying in the physical experiments because no precise measuring devices are available. A more dedicated calibration method by using precise measuring devices (e.g., cameras, laser interferometry, or supersonic distance sensors) should be developed to improve the accuracy of the robot, as well as the accuracy of material deposition.

The second thing to improve is software integration. In this research project, the realization of the whole AM workflow has not been integrated into a single, user-friendly program for product designers. Specifically, the slicing and tool-path generation are implemented in a standalone C++ based program, while the trajectory optimization and robot control are implemented based on the ROS framework. Although these two parts are bridged with the extended G-Code, it would be more convenient for designers by integrating the whole workflow as a single program. In addition, an easy-to-use user interface is also needed for end users. In future research, an end-to-end, user-friendly software solution should be developed that is able to cover the workflow from the design model to the final product in one program.

6.2.2 Surface Quality

Although a jerk-optimized motion planning algorithm is proposed for the robot to realize smooth motion to improve the fabrication quality, the fabrication quality still can not compete with that of a commercial FDM system. The main reasons are caused by the discretization error in the curved layer slicing algorithm and the inherent inaccuracy of the low-end articulated robotic arm.

To minimize the discretization error in the curved layer slicing algorithm, future research can use a tetrahedral mesh to produce an approximation of the input model instead of using voxel grids as in current work. Also, to diminish the staircase effect, the field-based method should take the constraint of surface covering into account.

On the hardware side, future research can develop a new AM system equipped with a high precision, five-axis, table tilting motion system. Because there will be no additional DOFs to optimize the motion smoothness and avoid gouging, a better scheme to compute continuous and optimized gouging-free printing orientation should also be developed.

6.2.3 Continuous-Fiber-Based Robot-Assisted AM

A major limitation of the work presented in Chapter 5 is that the process planning of embedding fibers and other functional components (such as electronics) are realized through a manual operation. With the increase of a part's complexity, automated process planning for non-planar placement of continuous-fiber types is required. To automate and optimize the process of direct deposition, future work could be done to develop such an extrusion head with an advanced control system.

Moreover, the direct deposition of carbon fiber tapes is still done manually in the experimental AM system. Future research should focus on developing an automatic rolling mechanism to work together with the robotic system for automating the process of carbon-fiber deposition.

Last but not least, the computational methods used to optimize fabrication sequence with continuous-fiber reinforcement need to be investigated in the future. The opportunity should be explored to optimize the accumulation field with the optimal stress distribution in the curved layer slicing algorithm.

6.3 Reflection on the process

In this study, four research cycles were presented under the unifying theme of developing a robot-assisted AM system that can overcome the problems designers face while using the FDM technology for AM. In the Research Cycle 1, a prototype system was developed to prove the concept of the advantage of rotational motion enriched AM. This exploration helped me to explore the insights into research questions and obtain hands-on experience. Also, the limitations in this research cycle formed the basis for the development of this research project.

The following three research cycles were sequential and parallel at the same time. The main themes of these three research cycles were different. Research Cycle 2 focused on slicing and tool-path generation in the 3D domain for AM. Research Cycle 3 focused on the hardware realization. Research Cycle 4 focused on the extension of the robot-assisted AM system by using the continuous-fiber material. Although each research cycle had a different focus, the development was iterative. For example, during the physical experiments conducted in Research Cycle 2, the need for an efficient and effective motion planning methods arose, thereby fueling the themes to be explored in the next research step.

6.4 Implications of this Research

The research described in this thesis opens the door to a new AM approach. It has been shown that the methods developed in this research successfully mitigate weaknesses in conventional AM by reducing the use of support structures and by enhancing the mechanical performance. This research not only provides the designers with more freedom to realize their creativity while using AM, but it also inspires AM service providers to develop a new type of AM machine to expand their product portfolios.

On the designer side, manufacturing the product in a support-free manner allows designers to create nearly any shape they can think of with no need for extra material and post-processing work. Hollowed parts comprise one particular product category for which the importance of support-free is particularly obvious. Hollowed parts contain voids to reduce weight and change mechanical performance. It is hard to use conventional AM to fabricate hollowed parts without support structures in interior voids, as supports cannot be removed from inside. Therefore, the system presented in this thesis can be useful to such product design.

Because the articulated robotic arm is designed for heavy, repetitive manufacturing work, it is definitely not the most commercial choice for being used as the portable AM platform for designers. On the AM service provider side, AM machine manufacturers can develop dedicated hardware to realize the same functions as a robot-assisted AM system with much less cost and more accuracy, thereby expanding the product portfolios for their company. Furthermore, this study also lays down a challenge for software developers to develop a robot-oriented CAD/CAM software to provide an end-to-end AM solution for robotic manufacturing systems with all the benefits discovered in this research.

Appendix

This appendix contains the description of an extended G-code file format for defining the tool-paths that can realize the robot-assisted AM. Detail format specification and example G-code files can be found below.

G-Code for Conventional AM

The G-code file for supporting robot-assisted AM is extended from the original RepRap G-code that has been widely used in 3D printing community. Here the most commonly used G-commands in RepRap G-code for conventional planar-layered AM are reviewed.

Example Code:

```
G92 E0
G28
G0 F600   X4.391  Y−8.398 Z1.500
G1 X3.650 Y−9.611 Z1.500  E1.026
```

The meaning of each command line is as follows.

- G92 E0: Reset extruder distance position.
- G28: Move to the home position.
- G0 is a command for rapid movement, where F600 specifies the speed as 600 (mm / min.) for the nozzle movement to the followed positions specified by Xnnn, Ynnn and Znnn (unit: mm).
- G1 stands for a linear movement to the specified position, where Ennn gives the amount of material to extrude between the starting point and ending point (e.g., 1.026mm of material will be extruded by the above code).

For the detailed explanation for all different commands and parameters, please refer to [120].

Extended G-code File Format

When fabricating a solid model by robot-assisted AM, not only the position but also the orientation of nozzle in the Euclidean coordinate system needs to be specified. In this extension, the orientation of nozzles is defined by the nXnnn, nYnnn and nZnnn flags. One example line of command with nozzle orientation is shown below.

```
...  ...
G1 X6.338  Y8.390  Z46.300  nX−0.094  nY0.127  nZ0.987  E4.324
...  ...
```

nX-0.094 nY0.127 nZ0.987 define the vector (-0.094, 0.127, 0.987) as the orientation of extrusion head (in the coordinate system of workpiece). All other flags of the RepRap G-code format are used in the same way.

Acknowledgments

First and foremost I would like to thank my promoter Prof. Charlie C.L. Wang. There is no enough word to express my thanks. You are the person who changes my life in every aspect. You introduce me to the world of research and I enjoy every minute. You let me realize my potential and keep me confident to face all difficulties in my research and daily life.

Next, I would like to thank my family. Lixin, you are my ambition and my best wife. I cannot make it without your mental support every day. I am so happy thinking about the time staying with you every time. Also, I would like to thank my parents for supporting every of my decisions. Although I have only spend less than a month with you in the last 4 years, I can still feel your love every day.

Thanks to all my TUDelft colleges who have made a great research atmosphere during my stay. First of all, my second promotor Prof. Jo M.P. Geraedts, you are really a nice guy that gives me good support for my research and thesis. Thanks to my best 'foreigner' friend, Rob, you are always willing to help me and also let me know the university and the country better. I would also like to thank Prof. Yu Song. Every time I worked until the closing alert ringing, I saw you were still at the office. This encourages me to work harder to be a good researcher like you. Furthermore, I would like to thank Guoxin for providing lots of help on the experiments and all the arguing we have. Thanks to other colleges for making it so much fun to work: Yabin, Zishun, Jun, Weiming, Wilemijn, Tim, Rado, Zjenja, Argun, Tessa, Sara, Martin, Adrie.

Many thanks to my other research collaborators to help me complete my papers. In particular, Chenming, you have played a crucial role in my research work. I enjoy all the collaboration with you. Thanks also to Prof. Sylvain Lefebvre. Although we have only met one time, your passion for research and additive manufacturing really impressed me, encouraging me a lot to do my research. Thanks to Prof. Jia Pan, Prof. Yong-jin Liu, Prof. Xiu-Jie Wang, Yajue, Bin Liu, Tianyu, Xiangjia, Zeyu, Shiqing, Allan Mok for providing help and research inspirations.

In the end, I would like to thank all my friends for enriching my life: Shuo Jin, Xiaoting, Yang Zheng, Kailun, Hong Yan, Chao Li, Wenjie Sun, Wenlong

Sun, Hao Jiang, Sheng Xu, Cheng Tan, Libo Zhou, Di Xu, Yarong Huai, Yang Yang, Zheyuan. I am always happy talking and hangout with you. I cannot list them all here and I would like to say sorry to the people that should have been mentioned.

Curriculum Vitæ

Chengkai Dai

04-09-1991	Born in Wuxi, China.

Education

2016–2020	PhD in Advanced Manufacturing Section, IDE Delft University of Technology, The Netherlands	
	Thesis:	Material Deposition in 3D Space – Additive Manufacturing Enriched by Rotational Motion
	Promotor:	Prof. dr. ir. Charlie C.L. Wang Prof. dr. ir. Jo M.P. Geraedts
2013–2014	Master of Science in Mechanical and Automation Engineering The Chinese University of Hong Kong, Hong Kong, China	
2009–2013	Undergraduate in Mechanical Engineering Nanjing Forestry University, Nanjing, China	

Research Experience

2014–2015 2018–2019	Research Assistant The Chinese University of Hong Kong, Hong Kong SAR, China

Publications

Journals

1. **Chengkai Dai**, Sylvain Lefebvre, Kai-Ming Yu, Jo M.P. Geraedts and Charlie C.L Wang, "Planning Jerk-Optimized Trajectory with Discrete-Time Constraints for Redundant Robots", *IEEE Transactions on Automation Science and Engineering*, accepted, 2020.
2. Chuhua Xian*, Dongjiu Zhang*, **Chengkai Dai**, and Charlie C. L. Wang, "Fast Generation of High Fidelity RGB-D Images by Deep-Learning with Adaptive Convolution", *IEEE Transactions on Automation Science and Engineering*, accepted, 2020 (* Joint First Authors)
3. Chenming Wu, **Chengkai Dai**, Guoxin Fang, Yong-Jin Liu and Charlie C.L. Wang, "General support-effective decomposition for multi-directional 3D printing", *IEEE Transactions on Automation Science and Engineering*, accepted, 2019
4. Chenming Wu, **Chengkai Dai**, Xiaoxi Gong, Yong-Jin Liu, Jun Wang, Xianfeng Gu, and Charlie C.L. Wang, "Energy-efficient coverage path planning for general terrain surfaces", *IEEE Robotics and Automation Letters*, vol.4, no.3, pp.2584-2591, 2019
5. **Chengkai Dai**, Charlie C.L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, Yong-Jin Liu, "Support-Free Volume Printing by Multi-Axis Motion", *ACM Transactions on Graphics (SIGGRAPH 2018)*, Vol. 37, No. 4, Article No.1, 2018
6. Shuo Jin, **Chengkai Dai**, Yang Liu, and Charlie C.L. Wang, "Motion imitation based on sparsely sampled correspondence", *ASME Journal of Computing and Information Science in Engineering*, Vol.17, No. 4, 2017

Conferences

1. Xiaoting Zhang, Guoxin Fang, **Chengkai Dai**, Jouke Verlinden, Jun Wu, Emily Whiting, and Charlie C.L. Wang, "Thermal-comfort design of personalized casts", *ACM Symposium on User Interface Software and Technology (UIST)*, pp.243-254, Quebec City, Canada, October 22-25, 2017
2. Chenming Wu*, **Chengkai Dai***, Guoxin Fang, Yong-Jin Liu, and Charlie C.L. Wang, "RoboFDM: a robotic system for support-free fabrication using FDM", *IEEE International Conference on Robotics and Automation (ICRA)*, pp.1175-1180, Singapore, May 29 - June 3, 2017 (* Joint First Authors)
3. Qianwen Chao, Jiangfan Yu, **Chengkai Dai**, Tiantian Xu, Li Zhang, Charlie C.L. Wang, and Xiaogang Jin, "Steering micro-robotic swarm by dynamic actuating fields", *IEEE International Conference on Robotics and Automation (ICRA)*, pp.5230-5235, Stockholm, Sweden, May 16-21, 2016
4. Wuyuan Xie, **Chengkai Dai**, and Charlie C.L. Wang, "Photometric stereo with near point lighting: A solution by mesh deformation", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, June 7-12, 2015

Book Chapters

1. Chi-Chung Li, **Chengkai Dai**, Wei-Hsin Liao, and Charlie C.L. Wang, "Towards Direct Deposition of Continuous-Fibers on Curved Surfaces", *Recent Advances in Additive Manufacturing*, Chapter 4, 2020

Bibliography

[1] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, *The status, challenges, and future of additive manufacturing in engineering,* Computer-Aided Design **69**, 65 (2015).

[2] H. Zhao, F. Gu, Q.-X. Huang, J. Garcia, Y. Chen, C. Tu, B. Benes, H. Zhang, D. Cohen-Or, and B. Chen, *Connected fermat spirals for layered fabrication,* ACM Transactions on Graphics **35**, 100:1 (2016).

[3] S. Kalpakjian, S. Schmid, and V. Sekar, *Manufacturing Engineering and Technology* (Pearson, 2013).

[4] 3D Hubs Manufacturing LLC., *3D printing Trends 2020*, Tech. Rep. (2020).

[5] X. Zhang, G. Fang, C. Dai, J. Verlinden, J. Wu, E. Whiting, and C. C. Wang, *Thermal-comfort design of personalized casts,* in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (2017) pp. 243—254.

[6] 3D Hubs Manufacturing LLC., *3D printing Trends Q1 2019*, Tech. Rep. (2019).

[7] ASTM, *Standard Terminology for Additive Manufacturing Technologies,* (2012).

[8] K. V. Wong and A. Hernandez, *A review of additive manufacturing, ISRN Mechanical Engineering*, (2012).

[9] A. Gebhardt, *Understanding Additive Manufacturing* (Hanser, 2011).

[10] E. Doubrovski, E. Tsai, D. Dikovsky, J. Geraedts, H. Herr, and N. Oxman, *Voxel-based fabrication through material property mapping,* Computer-Aided Design **60**, 313 (2015).

[11] M. Leary, *Design for Additive Manufacturing* (Elsevier, 2019).

[12] T. Yunlong and Z. Y. Fiona, *A survey of the design methods for additive manufacturing to improve functional performance, Rapid Prototyping Journal*, **22**, 569 (2016).

[13] *Optimal Design for Additive Manufacturing: Opportunities and Challenges*, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 9 (2011).

[14] M. Srivastava, S. Rathee, S. Maheshwari, and T. Kundra, *Additive Manufacturing: Fundamentals and Advancements* (Taylor & Francis, 2019).

[15] R. Ponche, J. Hascoet, O. Kerbrat, and P. Mognol, *A new global approach to design for additive manufacturing,* Virtual and Physical Prototyping **7**, 93 (2012).

[16] R. Ponche, O. Kerbrat, P. Mognol, and J.-Y. Hascoet, *A novel methodology of design for additive manufacturing applied to additive laser manufacturing process,* Robotics and Computer-Integrated Manufacturing **30**, 389 (2014).

[17] J. Etienne, N. Ray, D. Panozzo, S. Hornus, C. C. L. Wang, J. Martinez, S. McMains, M. Alexa, B. Wyvill, and S. Lefebvre, *CurviSlicer: Slightly curved slicing for 3-axis printers,* ACM Transactions on Graphics **38**, 81:1 (2019).

[18] J. Hartcher-O'Brien, J. Evers, and E. Tempelman, *Surface roughness of 3d printed materials: Comparing physical measurements and human perception,* Materials Today Communications **19**, 300 (2019).

[19] C. E. Duty, J. A. Failla, S. Kim, J. M. Lindahl, B. K. Post, L. J. Love, and V. Kunc, *Reducing mechanical anisotropy in extrusion-based printed parts,* in *Solid Freeform Fabrication Symposium* (2017).

[20] T. Zegard and G. H. Paulino, *Bridging topology optimization and additive manufacturing,* Structural and Multidisciplinary Optimization **53**, 175 (2016).

[21] W. Gao, Y. Zhang, D. C. Nazzetta, K. Ramani, and R. J. Cipra, *RevoMaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform,* in *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (2015) pp. 437–446.

[22] S. Keating and N. Oxman, *Compound fabrication: A multi-functional robotic platform for digital design and fabrication,* Robotics and Computer-Integrated Manufacturing **29**, 439 (2013).

[23] S. Mueller, S. Im, S. Gurevich, A. Teibrich, L. Pfisterer, F. Guimbretière, and P. Baudisch, *Wireprint: 3D printed previews for fast prototyping,* in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (2014) pp. 273–280.

[24] Y. Pan, C. Zhou, Y. Chen, and J. Partanen, *Multitool and multi-axis computer numerically controlled accumulation for fabricating conformal features on curved surfaces,* ASME Journal of Manufacturing Science and Engineering **136** (2014).

[25] X. Song, Y. Pan, and Y. Chen, *Development of a low-cost parallel kinematic machine for multidirectional additive manufacturing,* ASME Journal of Manufacturing Science and Engineering **137** (2015).

[26] K. Hu, S. Jin, and C. C. L. Wang, *Support slimming for single material based additive manufacturing,* Computer-Aided Design **65**, 1 (2015).

[27] K. Hu, X. Zhang, and C. C. L. Wang, *Direct computation of minimal rotation for support slimming,* in *Proceedings of IEEE International Conference on Automation Science and Engineering* (2015) pp. 936–941.

[28] P. Herholz, W. Matusik, and M. Alexa, *Approximating free-form geometry with height fields for manufacturing,* Computer Graphics Forum **34**, 239 (2015).

[29] R. Hu, H. Li, H. Zhang, and D. Cohen-Or, *Approximate pyramidal shape decomposition,* ACM Transactions on Graphics **33**, 213:1 (2014).

[30] C.J.Kruit, *A Novel Additive Manufacturing Approach Using a Multiple Degrees of Freedom Robotic Arm*, Master's thesis, Delft University of Technology (2013).

[31] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, *Mean curvature skeletons,* Computer Graphics Forum **31**, 1735 (2012).

[32] L. Shapira, A. Shamir, and D. Cohen-Or, *Consistent mesh partitioning and skeletonisation using the shape diameter function,* Visual Computer **24**, 249 (2008).

[33] L. Luo, I. Baran, S. Rusinkiewicz, and W. Matusik, *Chopper: Partitioning models into 3d-printable parts,* ACM Transactions on Graphics **31**, 129:1 (2012).

[34] J. J. Kuffner and S. M. Lavalle, *RRT-connect: An efficient approach to single-query path planning,* in *Proceedings of IEEE International Conference on Robotics and Automation* (2000) pp. 995–1001.

[35] I. Gibson, D. W. Rosen, and B. Stucker, *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing* (Springer, 2015).

[36] D. Chakraborty, B. Aneesh Reddy, and A. Roy Choudhury, *Extruder path generation for curved layer fused deposition modeling,* Computer-Aided Design **40**, 235 (2008).

[37] T. Llewellyn-Jones, A. Robert, and T. Richard, *Curved layer fused filament fabrication using automated tool-path generation,* 3D Printing and Additive Manufacturing **3**, 236 (2016).

[38] B. Ezair, S. Fuhrmann, and G. Elber, *Volumetric covering print-paths for additive manufacturing of 3d models,* Computer-Aided Design **100**, 1 (2018).

[39] H. Peng, R. Wu, S. Marschner, and F. Guimbretière, *On-the-fly print: Incremental printing while modelling,* in *Proceedings of the CHI Conference on Human Factors in Computing Systems* (ACM, 2016) pp. 887–896.

[40] R. Wu, H. Peng, F. Guimbretière, and S. Marschner, *Printing arbitrary meshes with a 5DOF wireframe printer,* ACM Transactions on Graphics **35**, 101:1 (2016).

[41] Y. Huang, J. Zhang, X. Hu, G. Song, Z. Liu, L. Yu, and L. Liu, *FrameFab: Robotic fabrication of frame shapes,* ACM Transactions on Graphics **35**, 224:1 (2016).

[42] G. Marsh, *Automating aerospace composites production with fibre placement,* Reinforced Plastics **55**, 32 (2011).

[43] P. Song, B. Deng, Z. Wang, Z. Dong, W. Li, C.-W. Fu, and L. Liu, *Cofifab: Coarse-to-fine fabrication of large 3d objects,* ACM Transactions on Graphics **35**, 45:1 (2016).

[44] C. Wu, C. Dai, G. Fang, Y. Liu, and C. C. L. Wang, *Robofdm: A robotic system for support-free fabrication using fdm,* in *Proceedings of IEEE International Conference on Robotics and Automation* (2017) pp. 1175–1180.

[45] G. Elber, *Accessibility in 5-axis milling environment,* Computer-Aided Design **26**, 796 (1994).

[46] O. Ilushin, G. Elber, D. Halperin, R. Wein, and M.-S. Kim, *Precise global collision detection in multi-axis nc-machining,* Computer-Aided Design **37**, 909 (2005).

[47] N. Wang and K. Tang, *Automatic generation of gouge-free and angular-velocity-compliant five-axis toolpath,* Computer-Aided Design **39**, 841 (2007).

[48] Y.-J. Kim, G. Elber, M. Bartoň, and H. Pottmann, *Precise gouging-free tool orientations for 5-axis cnc machining,* Computer-Aided Design **58**, 220 (2015).

[49] M. Liu, Y.-s. Liu, and K. Ramani, *Computing global visibility maps for regions on the boundaries of polyhedra using minkowski sums,* Computer-Aided Design **41**, 668 (2009).

[50] X. Zhang, K.-C. Chan, C. C. L. Wang, K.-C. Wong, and S.-M. Kumta, *Computing stable contact interface for customized surgical jigs,* in *Proceedings of IEEE International Conference on Robotics and Automation* (2015) pp. 6160–6166.

[51] J. Pan, S. Chitta, and D. Manocha, *FCL: A general purpose library for collision and proximity queries,* in *Proceedings of IEEE International Conference on Robotics and Automation* (2012) pp. 3859–3866.

[52] T. Ju, F. Losasso, S. Schaefer, and J. Warren, *Dual contouring of hermite data,* ACM Transactions on Graphics **21**, 339 (2002).

[53] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm,* in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987) pp. 163–169.

[54] Q. Zhou, E. Grinspun, D. Zorin, and A. Jacobson, *Mesh arrangements for solid geometry,* ACM Transactions on Graphics **35**, 39:1 (2016).

[55] C. Xu, T. Y. Wang, Y.-J. Liu, L. Liu, and Y. He, *Fast wavefront propagation (FWP) for computing exact geodesic distances on meshes,* IEEE Transactions on Visualization & Computer Graphics **21**, 822 (2015).

[56] S. Gottschalk, M. C. Lin, and D. Manocha, *OBBTree: A hierarchical structure for rapid interference detection,* in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996) pp. 171–180.

[57] H.-C. Song, N. Ray, D. Sokolov, and S. Lefebvre, *Anti-aliasing for fused filament deposition,* Computer-Aided Design **89**, 25 (2017).

[58] C. Dai, C. C. L. Wang, C. Wu, S. Lefebvre, G. Fang, and Y.-J. Liu, *Support-free volume printing by multi-axis motion,* ACM Transactions on Graphics **37**, 134:1 (2018).

[59] M. Ibrahim, N. Isa, D. N. Saude, and M. Ibrahim, *Verification of feed rate effects on filament extrusion for freeform fabrication,* Journal of Engineering and Applied Sciences **11**, 6556 (2016).

[60] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, *Motion planning with sequential convex optimization and convex collision checking,* The International Journal of Robotics Research **33**, 1251 (2014).

[61] D. P. Martin, J. Baillieul, and J. M. Hollerbach, *Resolution of kinematic redundancy using optimization techniques,* IEEE Transactions on Robotics and Automation **5**, 529 (1989).

[62] B. Siciliano, *Kinematic control of redundant robot manipulators: A tutorial,* Journal of Intelligent and Robotic Systems **3**, 201 (1990).

[63] P. Cheng and S. M. LaValle, *Resolution complete rapidly-exploring random trees,* in *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1 (2002) pp. 267–272.

[64] S. Karaman and E. Frazzoli, *Sampling-based algorithms for optimal motion planning,* International Journal of Robotics Research **30**, 846 (2011).

[65] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, 1st ed. (Addison-Wesley, 1990).

[66] T. Yoshikawa, *Basic optimization methods of redundant manipulators,* Laboratory Robotics and Automation **8**, 49 (1996).

[67] L. Huo and L. Baron, *The joint-limits and singularity avoidance in robotic welding,* Industrial Robot: An International Journal **35** (2008), 10.1108/01439910810893626.

[68] S. Seereeram and J. T. Wen, *A global approach to path planning for redundant manipulators,* in *Proceedings of IEEE International Conference on Robotics and Automation* (1993) pp. 283–288 vol.2.

[69] S. E. Khadem and R. V. Dubey, *A global cartesian space obstacle avoidance scheme for redundant manipulators,* Optimal Control Applications and Methods **12**, 279 (1991).

[70] J. Léger and J. Angeles, *Off-line programming of six-axis robots for optimum five-dimensional tasks,* Mechanism and Machine Theory **100**, 155 (2016).

[71] R. Dubey and J. Y. S. Luh, *Redundant robot control using task based performance measures,* Journal of Robotic Systems **5**, 409 (1988).

[72] P. Freeman, *Minimum jerk trajectory planning for trajectory constrained redundant robots,* Ph.D. Dissertation (2012).

[73] Z. Zhang, S. Chen, X. Zhu, and Z. Yan, *Two hybrid end-effector posture-maintaining and obstacle-limits avoidance schemes for redundant robot manipulators,* IEEE Transactions on Industrial Informatics , 1 (2019).

[74] L. E. Kavraki, P. Svestka, J. . Latombe, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces,* IEEE Transactions on Robotics and Automation **12**, 566 (1996).

[75] S. M. Lavalle, J. J. Kuffner, and Jr., *Rapidly-exploring random trees: Progress and prospects,* in *Algorithmic and Computational Robotics: New Directions* (2000) pp. 293–308.

[76] M. Stilman, *Global manipulation planning in robot joint space with task constraints,* IEEE Transactions on Robotics **26**, 576 (2010).

[77] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, *Manipulation planning on constraint manifolds,* in *Proceedings of IEEE International Conference on Robotics and Automation* (2009) pp. 625–632.

[78] Q. Qiu and Q. Cao, *Task constrained motion planning for 7-degree of freedom manipulators with parameterized submanifolds,* Industrial Robot: the international journal of robotics research and application **45**, 363 (2018).

[79] J. Pan and D. Manocha, *Efficient configuration space construction and optimization for motion planning,* Engineering **1**, 046 (2015).

[80] J. Pan, X. Zhang, and D. Manocha, *Efficient penetration depth approximation using active learning,* ACM Transactions on Graphics **32**, 191:1 (2013).

[81] N. Das, N. Gupta, and M. Yip, *Fastron: An online learning-based model and active learning strategy for proxy collision detection,* in *Proceedings of 1st Annual Conference on Robot Learning* (2017) pp. 496–504.

[82] J. Huh and D. D. Lee, *Learning high-dimensional mixture models for fast collision detection in rapidly-exploring random trees,* in *Proceedings of IEEE International Conference on Robotics and Automation* (2016) pp. 63–69.

[83] J. Pan and D. Manocha, *Fast probabilistic collision checking for sampling-based motion planning using locality-sensitive hashing,* The International Journal of Robotics Research **35**, 1477 (2016).

[84] I. Garcia, J. D. Martin-Guerrero, E. Soria-Olivas, R. J. Martinez, S. Rueda, and R. Magdalena, *A neural network approach for real-time collision detection,* in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5 (2002) pp. 1–5.

[85] S. S. M. Salehian, N. Figueroa, and A. Billard, *A unified framework for coordinated multi-arm motion planning,* The International Journal of Robotics Research **37**, 1205 (2018).

[86] C. J. C. Burges, *A tutorial on support vector machines for pattern recognition,* Data Mining and Knowledge Discovery **2**, 121 (1998).

[87] B. Scholkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, *Comparing support vector machines with gaussian kernels to radial basis function classifiers,* IEEE Transactions on Signal Processing **45**, 2758 (1997).

[88] L. Chen, Y. Li, and K. Tang, *Variable-depth multi-pass tool path generation on mesh surfaces,* International Journal of Advanced Manufacturing Technology **95**, 2169 (2018).

[89] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines,* ACM Transactions on Intelligent Systems and Technology **2**, 27:1 (2011).

[90] Stratasys, *3D printing with FDM and Nylon 12,* `http://www.stratasys.com/materials/fdm/nylon` (2015).

[91] Z. Qin, B. G. Compton, J. A. Lewis, and M. J. Buehler, *Structural optimization of 3D-printed synthetic spider webs for high strength,* Nature Communications **6**, 7038 (2015).

[92] F. Lederle, F. Meyer, G.-P. Brunotte, C. Kaldun, and E. G. Hübner, *Improved mechanical properties of 3D-printed parts by fused deposition modeling processed under the exclusion of oxygen,* Progress in Additive Manufacturing **1**, 3 (2016).

[93] J. T. Belter and A. M. Dollar, *Strengthening of 3D printed fused deposition manufactured parts using the fill compositing technique,* PLoS ONE **10**, 1 (2015).

[94] B. G. Compton and J. A. Lewis, *3D-printing of lightweight cellular composites,* Advanced Materials **26**, 5930 (2014).

[95] S. Christ, M. Schnabel, E. Vorndran, J. Groll, and U. Gbureck, *Fiber reinforcement during 3D printing,* Materials Letters **139**, 165 (2015).

[96] H. L. Tekinalp, V. Kunc, G. M. Velez-Garcia, C. E. Duty, L. J. Love, A. K. Naskar, C. A. Blue, and S. Ozcan, *Highly oriented carbon fiber–polymer composites via additive manufacturing,* Composites Science and Technology **105**, 144 (2014).

[97] F. Ning, W. Cong, J. Qiu, J. Wei, and S. Wang, *Additive manufacturing of carbon fiber reinforced thermoplastic composites using fused deposition modeling,* Composites Part B: Engineering **80**, 369 (2015).

[98] T. M. Llewellyn-Jones, B. W. Drinkwater, and R. S. Trask, *3D printed components with ultrasonically arranged microscale structure,* Smart Materials and Structures **25**, 02LT01 (2016).

[99] H. Brooks and S. Molony, *Design and evaluation of additively manufactured parts with three dimensional continuous fibre reinforcement,* Materials & Design **90**, 276 (2016).

[100] J. M. Canela, I. B. Corral, L. Bade, P. Hackney, I. Shyha, and M. Birkett, *Investigation into the development of an additive manufacturing technique for the production of fibre composite products,* Procedia Engineering **132**, 86 (2015).

[101] T.-H. Kwok, Y. Li, and Y. Chen, *A structural topology design method based on principal stress line,* Computer-Aided Design **80**, 19 (2016).

[102] K.-M. M. Tam and C. T. Mueller, *Additive manufacturing along principal stress lines,* 3D Printing and Additive Manufacturing **4** (2017).

[103] Y. Chen, C. Zhou, and J. Lao, *A layerless additive manufacturing process based on CNC accumulation,* Rapid Prototyping Journal **17**, 218 (2011).

[104] H. Prüß and T. Vietor, *Design for fiber-reinforced additive manufacturing,* Journal of Mechanical Design **137**, MD (2015).

[105] Z. August, G. Ostrander, J. Michasiow, and D. Hauber, *Recent developments in automated fiber placement of thermoplastic composites,* SAMPE Journal **50** (2014).

[106] Markforged, *The Mark Two industrial strength 3D printer,* `https://markforged.com/mark-two/` (2016).

[107] Markforged Material, *Materials offered by Markforged,* `http://markforged.com/order-materials/` (2016).

[108] EasyComposite, `http://www.easycomposites.co.uk/` (2015).

[109] G. T. Mark and A. S. Gozdz, *Three dimensional printer with composite filament fabrication,* US Patent 9156205 B2 (2015).

[110] R. Matsuzaki, M. Ueda, M. Namiki, T.-K. Jeong, H. Asahara, K. Horiguchi, T. Nakamura, A. Todoroki, and Y. Hirano, *Three-dimensional printing of continuous-fiber composites by in-nozzle impregnation,* Scientific Reports **6** (2016).

[111] X. Tian, T. Liu, C. Yang, Q. Wang, and D. Li, *Interface and performance of 3D printed continuous carbon fiber reinforced PLA composites,* Composites Part A: Applied Science and Manufacturing **88**, 198 (2016).

[112] N. Li, Y. Li, and S. Liu, *Rapid prototyping of continuous carbon fiber reinforced polylactic acid composites by 3D printing,* Journal of Materials Processing Technology **238**, 218 (2016).

[113] S. Singamneni, A. Roychoudhury, O. Diegel, and B. Huang, *Modeling and evaluation of curved layer fused deposition,* Journal of Materials Processing Technology **212**, 27 (2012).

[114] B. Huang and S. B. Singamneni, *Curved layer adaptive slicing (CLAS) for fused deposition modelling,* Rapid Prototyping Journal **21**, 354 (2015).

[115] Y. Chen and C. C. L. Wang, *Uniform offsetting of polygonal model based on layered depth-normal images,* Computer-Aided Design **43**, 31 (2011).

[116] Y. Song, R. A. Boekraad, L. Roussos, A. Kooijman, C. C. Wang, and J. M. Geraedts, *3D printed electronics: opportunities and challenges from case studies,* in *Proceedings of ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* (2017).

[117] Thermaltronics, *Series K blade tip,* `http://www.thermaltronics.com/k_series.php`.

[118] R. Pitchumani, J. W. Gillespie Jr., and M. A. Lamontia, *Design and optimization of a thermoplastic tow-placement process with in-situ consolidation,* Journal of Composite Materials **31**, 244 (1997).

[119] J. Meyer, *Tool Point Calibration,* `https://github.com/Jmeyer1292/tool_point_calibration` (2016).

[120] RepRap, *G-code,* `https://reprap.org/wiki/G-code`.