# Radial Slicing for Helical-Shaped Advanced Manufacturing Applications

**Nuwan Munasinghe · Gavin Paul**

Received: date / Accepted: date

**Abstract** The fourth industrial revolution (Industry 4.0) is transforming industries all around the world focusing on areas including advanced robotics and automation, sensor technology and data analytics. The authors are involved in a project developing a multi-robot material extrusion 3D printer to print a Gravity Separation Spiral (GSS), an instrument used in the mining industry to separate mineral slurry into different density components. Compared to traditional mould-based manufacturing, this new additive manufacturing method will significantly reduce manufacturing tooling costs, improve the customisation to enable the production of bespoke GSS that each process different minerals, and reduce worker exposure to hazardous materials. Slicing and printing large scale helical objects in conventional horizontal layer addition would result in an unreasonable amount of waste material from support structures, and poor surface quality due to step-wise bumps. This paper presents a novel slicing algorithm using concentric vertical ray lines to slice objects radially, enabling layers to be deposited progressively in the same fashion. This method can be applied in large scale additive manufacturing where objects are printed by a robot in a radial direction, which is different from layered vertical printing in conventional additive systems. An example GSS is sliced to generate motion plans for a print head affixed to the end effector of a robot arm. Then through simulations, it is shown how a robot's expected manipulability measure can be used to predict and ensure the successful completion of the print.

**Keywords** Advanced manufacturing · Additive manufacturing · 3D printing · Slicing algorithm · Manipulability

## 1 Introduction

Additive manufacturing (AM), commonly known as 3D printing, is the process of depositing material layer by layer to form a physical realisation of a 3D computer model, as opposed to subtractive and formative manufacturing methods [17]. AM continues to evolve from just a useful rapid prototyping tool to one where final end-product manufacturing is possible [36]. Therefore, different industries such as construction and medicine are increasingly embracing this technology [30,8,4]. The mining industry in Australia is looking to integrate AM to overcome inherent drawbacks in traditional manufacturing. To this end, a research and development project is underway to create a material extrusion 3D printer to manufacture Gravity Separation Spirals (GSS), which are used to separate minerals from the slurry. The slurry is pumped to the top of the spirals, and gravity, combined with the contour of the spiral, effectively separates the slurry into its constituent minerals, according to their specific gravities. The profile of the spiral is designed based upon the target mineral and the composition of the ore body. Therefore, each customer has a different set of requirements based upon the target mineral type, which would be best served by customising the contour and shape of each spiral. However, it is a difficult and uneconomical task to produce bespoke spirals with a traditional mould-based manufacturing

This research is supported by UTS, in particular, Rapido; The Commonwealth of Australia's Department of Industry, Innovation and Science (Innovative Manufacturing CRC Ltd); and Downer, via its subsidiary Mineral Technologies.

Nuwan Munasinghe[1], Gavin Paul [2]
Centre for Autonomous Systems
University of Technology Sydney
Australia
E-mail: {[1]Nuwan.Munasinghe,[2]Gavin.Paul}@uts.edu.au

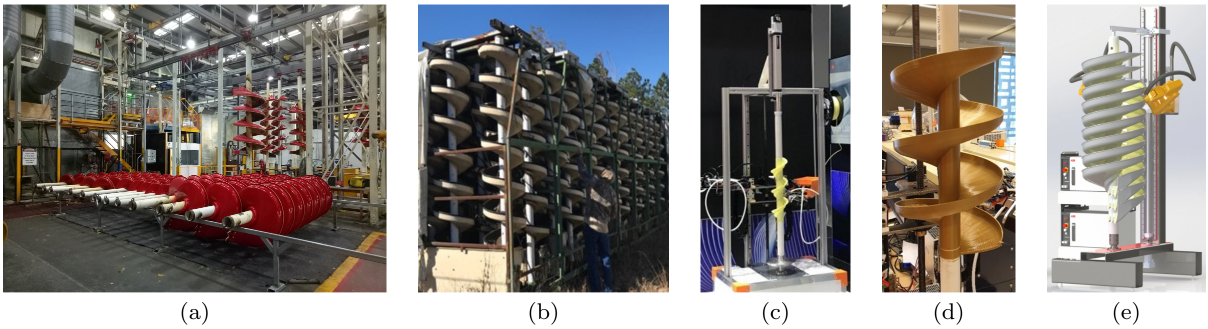


Fig. 1: a) GSS manufacturing factory; b) Bank of GSS installed in the field; c) One-third-of scale printer; d) Printed spiral using small scale printer e) Full-scale printer under development

technique, due to the time and cost to recreating new moulds, and the high manual labour input. Additionally, the current manufacturing method, shown in Fig. 1a, requires human labourers to use hazardous materials. Consequently, there is a potential health risk associated with exposing workers to such materials. These spirals will normally operate as large spiral banks as shown in Fig. 1b. The Rapido team at the University of Technology Sydney, in collaboration with Downer's Mineral Technologies and supported by the Australian Innovative Manufacturing CRC, have developed a 1:3 scale 3D printer (Fig. 1c) which can print spirals (Fig. 1d), with Fig. 1e shows an illustration of the full-scale printer design under development [2].

Alongside the development of a 3D printer to print these GSS, researchers are concurrently researching and developing sensors that can be embedded inside the spirals, or mounted externally to the spiral, to measure various parameters like wear and strain [23,26,24,25]. Ultimately, the aim of embedding sensors to the spiral is to monitor operating conditions of the spirals that are installed in various remote parts of countries such as USA, India, South Africa, Brazil, and Australia. By doing so, it is possible to reduce the need for technicians to travel to these remote locations to troubleshoot issues. Additionally, by having a detailed history of operation usage that is based on real-time data from embedded sensors, fault diagnosis and operational efficiency can be improved significantly. To measure conditions remotely, different sensors have been developed and certain key sensors are 3D printable. Therefore, the authors introduced the term "printability" in 3D printing to indicate the amount of dexterity available to print in a given location in space and with a variety of velocities and orientations. This measure is used to identify areas where an increased degree of dexterity occurs, with the goal to enable the inline printing of intricate, embedded sensors. In this paper, the authors propose manipulability as a measure of printability, since it directly relates to the robot dexterity, and can be used to predict print failures caused by the robot's singularities. The manipulability measure is mathematically defined in Section 2.3.2. Authors also developed a method to maximise the manipulability and improve the overall dexterity in the print to make it easy for the inline print of sensors and improve the accuracy of the print.

Traditionally, there have been two types of popular slicing methods used in AM: direct slicing and STL-based slicing. Different CAD programs have different data formats. Direct slicing has the advantage of being derived from the original CAD model hence avoiding approximations that can adversely affect robustness [16]. Cao et al. [5] suggested a direct slicing algorithm that slices solid models in AutoCAD. Chang developed a direct slicing method to slice models from PowerSHAPE CAD software [7]. The main disadvantage of direct slicing is that it is not guaranteed to be compatible across different CAD software, or it can only be used with specific CAD software for a particular machine [9]. Therefore, STL-based slicing is commonly used in AM [16].

In STL-based slicing, the first step is to create a 3D model of the object using a CAD program and export it to a stereolithography or Standard Triangulation Language (STL) file. In contrast to direct slicing where slicing has been done in different CAD systems with their own data formats, in STL-based slicing, different CAD programs will export 3D models to a standardised STL file format and a program called slicer will slice it. STL files approximate the surface of the object with a large number of small triangular facets - also known as tessellation. STL files have been adopted as the de facto industry standard for AM because they are simple and robust for tessellation algorithms. Additionally, it makes the AM process independent from the CAD modelling methods. Because of this, STL files are used in various 3D printing methods like SLA (stereolithography apparatus), SLS (selective laser sintering), FDM (fused deposition modelling) and LOM (laminated object manufacturing) [41]. The exported STL file is sliced layer-by-layer since the AM printers print them in layers. In general, STL-based slicing will intersect STL files with a set of horizontal planes, and each will gives a piecewise linear contours of a slice. Next, these contours will be filled with a selected infill, and finally, a tool-path method will be applied in order to produce the printer's machine instructions [18].

Various slicing methods have been proposed for slicing STL files. Uniform slicing methods create layers with constant thickness, while adaptive slicing methods will create layers that have varying levels of thickness [18,39]. A simple uniform parallel slicing algorithm was developed by Kirschman et al. [20], which will intersect all triangles in plane Z and connect line segments into closed polygons for each slice top to bottom [39]. Adaptive slicing will generate slices with inconsistent spacing, which can be programmatically varied based upon parameters such as geometry and capacity. The advantage of this method is the reduction in build time and improvement in surface finish. Yang et al. [37] used an adaptive slicing algorithm to slice a point cloud. This method is based on adaptive slicing of moving least squares surfaces. Hybrid slicing is a combination of adaptive slicing and direct slicing [33]. All these methods rely on the intersections of planes and triangles to construct contour geometry. These plane and triangle intersections can cause singularity-case problems, such as isolated points and dangling edges [19].

The capability of a robotic manipulator to execute various tasks depends on the kinematic model, its location in physical space and environmental restrictions. Reachability and manipulability information can be used to design the process to obtain the correct desired performance of the task [29]. A robot workspace can be represented in a capability map, indicating how the robot manipulator can move within the surrounding space. A capability map will show places that can be easily reached by the manipulator. By using these maps, it is possible to either move objects where versatile manipulation is possible or move the manipulator to a different joint configuration so as to enable optimal, or at least, a higher degree of manipulation [3]. Researchers have applied dexterity measures for specific tasks such as an assembly task related to a planar robot arm [32], or to adaptively modify planar 3D-printing robot plans [34]. In other manufacturing processors like cutting, researchers have analysed the machinability effects of milling convex and concave surfaces [15,14].

The aforementioned slicing methods cannot be applied directly for this project's bespoke 3D printer since the printing operation is executed around a rotating centre column using robotic arms. Gang et al. [40] proposed a non-planar slicing method for stationery printing, which differs from normal slicing algorithms that slice in the horizontal plane [1,27,12]. However, this method is developed for printing small scale objects and furthermore, in our application, printing must occur about a non-stationary rotating column. Klosterman et al. [21] developed a curved layer laminated object manufacturing (LOM) method that increases speed-reducing waste, but this is a special process of LOM and the algorithm is unsuitable for other AM methods. Chakraborty et al. [6] presented a path generation model based on the parametric surface for AM adapting curved layer accumulation designed for thin parts. Printing a GSS for this project in the horizontal direction would require an unreasonable amount of support structure. It would also introduce step-wise bumps on the spiral surface that are unacceptable since this undesirable roughness affects a slurry's surface adhesion and thus the GSS's ability to perform appropriate separation. Additionally, printing needs to occur normal to the axis of the spiral. However, unlike traditional horizontal slicing algorithms, there is no single horizontal or vertical plane that slicing can be applied in. Thus, printing should occur in concentric cylindrical surfaces parallel to the spiral's central axis, and it should ideally incorporate the limitations of the robot. This paper presents a novel radial STL slicing algorithm that achieves this goal. The specific contributions are:

- A radial STL slicing algorithm to generate tool paths for printing helical shapes onto a central column without supports.
- Using manipulability measure to evaluate the capacity of a robot arm to execute tool paths in a robot-based printer.

The remainder of the paper is organised as follows. Firstly, Section 2 provides a mathematical background and theory related to the work. Section 3 provides results of the simulation, Section 4 provides a discussion about results and finally, Section 5 concludes the paper.

## 2 Methodology

### 2.1 Structure of the STL Files

STL files contain a list of triangle facet data that defines the surface model. A facet is uniquely identified by using a unit normal and three vertices. Each vertex is defined by three coordinates in $\mathbb{R}^3$, and each facet identifies the boundary between the exterior and the interior of the 3D object [18]. These triangles satisfy the following five conditions: (i) each edge common at

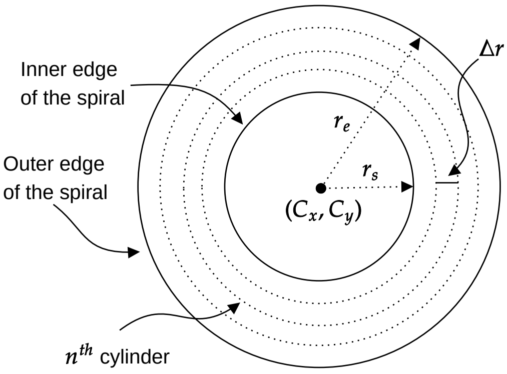


Fig. 2: Generated cylinders

most two triangles; (ii) a vertex can be shared by any number of triangles; (iii) each triangle has at least a single point in common with another triangle; (iv) if there is a common vertex with the second triangle, it is also a vertex of the second triangle; (v) triangles do not intersect with the interior of any other triangles [35].

## 2.2 Slicing Algorithm

### *2.2.1 Generation of Ray Lines*

Figure 2 shows a top-down view of the spiral. A number of cylinders, $N$ are calculated using (1) where user-defined limits of minimum radius, $r_s$ and maximum radius, $r_e$ limit where slicing should occur. The gap between two concentric cylinders, $\Delta r$ is also user-defined and $(C_x,\ C_y)$ is the centre of the spiral.

$$N \ = \ (r_e \ - \ r_s)/\Delta r \tag{1}$$

The length along the circumference of the cylinder, $\Delta C$ is a user-defined value. A point in the $n^{th}$ cylinder, $(x_0, y_0, z)$ is defined in (2) and (3) where $z$ is an arbitrary constant value. The total number of points around the circumference, $K$ which are equally spaced by $\Delta C$, is calculated using (5). For the each cylinder, a set of vertical lines, $L_{n,k}$ parallel to the spiral axis going through the point, $(x_k, y_k)$ are calculated as per (9) for the $n^{th}$ cylinder. These are shown in Fig. 3 where $n = 1 \ldots N$ and for the $k^{th}$ point where $k = 1 \ldots K$.

$$x_0 = C_x + r_s + n \cdot \Delta r \tag{2}$$
$$y_0 = C_y \tag{3}$$

$$r_n = |x_0 - C_x| \tag{4}$$
$$K = \lfloor 2 \cdot \pi \cdot r_n / \Delta C \rfloor \tag{5}$$
$$\Delta\theta = k \cdot \Delta C / r_n \tag{6}$$

$$x_k = (x_0 - C_x) \cdot \cos(\Delta\theta) + (y_0 - C_y) \cdot \sin(\Delta\theta) + C_x \tag{7}$$

$$y_k = -(x_0 - C_x) \cdot \sin(\Delta\theta) + (y_0 - C_y) \cdot \cos(\Delta\theta) + C_y \tag{8}$$

$$L_{n,k} =< x_k, y_k, z > +t < 0, 0, 1 > where\ t \in \mathbb{R} \tag{9}$$

### *2.2.2 Calculation of Intersections Between Rays and Model*

For each ray line $L_{n,k}$, the intersection points between the STL model and the lines are calculated. Since the model is composed of a set of triangles, finding triangles that a line intersects and calculating the intersection point between the line and triangles gives all the intersection points between a line and the model. To improve efficiency, a heuristic has been used. These user-defined thresholds $T_x$ and $T_y$ define a bounding box, as shown

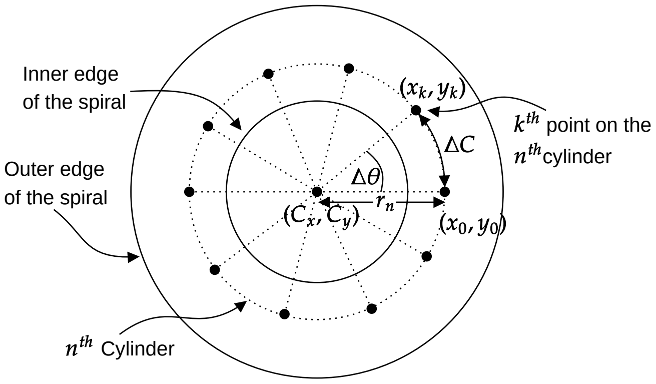


Fig. 3: Generated points for ray lines

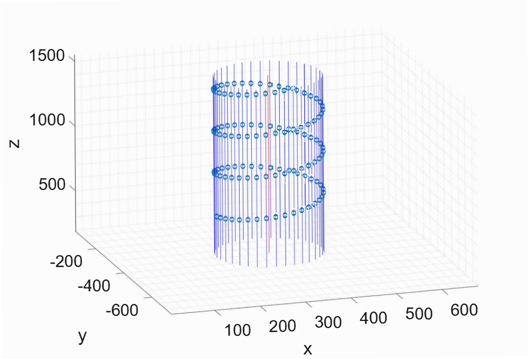


Fig. 4: Lines and intermediate points in a single cylinder

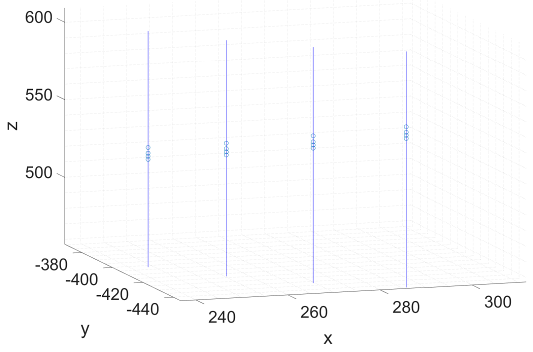


Fig. 5: Closer view of intermediate points

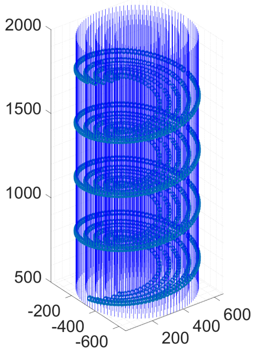


Fig. 6: Lines and intermediate points of all cylinders

in (10) to (13). The search is limited to only consider triangles within this bounding box when calculating the line-plane intersection points.

$$x^{+} = x_k + \ T_x \tag{10}$$
$$x^{-} = x_k - \ T_x \tag{11}$$
$$y^{+} = y_k + \ T_y \tag{12}$$
$$y^{-} = y_k - \ T_y \tag{13}$$

### *2.2.3 Generation of Intermediate Points*

After determining the intersection points between the lines and the model, those points are sorted by their Z height (i.e. along the Z-axis). In this step, the objective is to create a set of intermediate points with a user-defined layer height, $\Delta h$.

Since the ray lines are going through triangles, and the model is closed, there will be an even number of intersection points. Therefore, the point with the lowest Z value will be at the bottom surface and the next point will be at the top surface. This intermediate space should be filled during printing (i.e. the infill). A solid infill is desired by this project for various practical reasons. Given this assumption, the intermediate points are calculated as follows. Let the bottom and top point's X, Y and Z coordinates be $(X_b, Y_b, Z_b)$, and $(X_t, Y_t, Z_t)$, respectively. The number of intermediate points is $m$ and depends upon the thickness. Let the $i^{th}$ intermediate point, $P_i$ and the X,Y and Z values of that intermediate point be $(P_{i,x}, P_{i,y}, P_{i,z})$ as calculated in (15) to (17). The X and Y values in the intermediate points are equal to the X and Y values in the bottom and top points as shown in (15) and (16). Figure 4 shows these intermediate points in the overall view for a single cylinder and Fig. 5 shows a closer view of the same data, such that the individual points are clear. Figure 6 shows the outcome when these calculations are applied to all cylinders.

$$m = \lfloor (Z_t - Z_b)/\Delta h \rfloor \tag{14}$$
$$P_{i,x} = X_b = X_t \tag{15}$$
$$P_{i,y} = Y_b = Y_t \tag{16}$$
$$P_{i,z} = Z_b + i \cdot \Delta h \ where \ i = 1 \ldots m \tag{17}$$

### *2.2.4 Trajectory Creation*

Once a set of points, $S$, is generated, which includes the intermediate points and the top and the bottom points for the whole cylinder, it is necessary to create a tooltip trajectory. Trajectories are the sequence of points the tooltip of the robot should follow. The set of trajectories according to the points are created for points in each cylinder. In order to do that, for each cylinder, the lowest cluster of points are identified. The point with the lowest Z value, $M$ is selected. Consider X and Y values of $M$ are $M_x$ and $M_y$. The set of points, $N$ which has the same X and Y values (i.e. they are vertically above $M$ as calculated in (18)), are then sorted based upon their Z values. Consider a point $p_i$ that has a lower Z value, $z_i$ to $p_{i+1}$. The lowest cluster of points, $L_1$, which lie within the model are identified with (19). To create the set $L_1$, the difference between the Z values of $p_{i+1}$ and $p_i$ is compared with $\Delta h$. If it is less than $\Delta h$, it is added to the set. Once the condition is no longer satisfied the cluster is removed from the set $N$ as shown in (20). The same method is then applied to the remaining points to create the next cluster, and the process continues until $N = \emptyset$.

$$N = \{p \mid x = M_x, y = M_y, p \in S\} \tag{18}$$
$$L_1 = \{p \mid (z_{i+1} - z_i) < \Delta h\} \tag{19}$$
$$N = N - L_1 \tag{20}$$

Fig. 7 shows how sequences, or connections between points in each cluster, are created depending on the

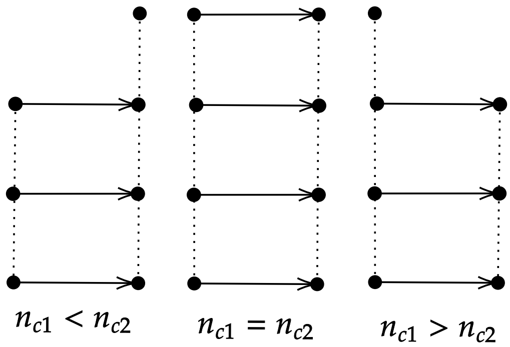


Fig. 7: Different methods of connecting clusters

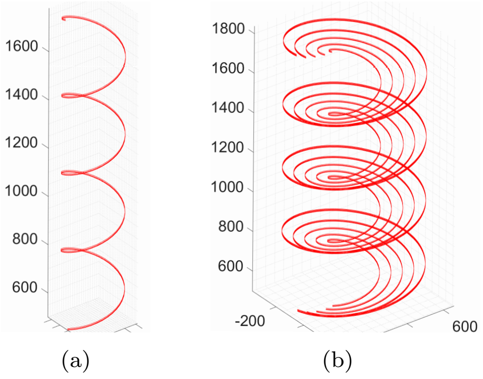
number of points in each cluster. An arrow shows the direction of the tool tip travel and $n_{c1}$ and $n_{c2}$ represents the number of points in each cluster. After connecting this cluster, this process continues for clusters in sequence $(L_1, L_2, L_3, \cdots)$ until $N = \emptyset$. Figure 8a shows trajectories generated for a single spiral and Fig. 8b shows trajectories generated for multiple cylinders.

## 2.3 Printing Process Simulation

A simulation environment shown in Figure 9 was developed in Matlab according to the actual physical dimensions of the printer. The dimensions of the industrial robot (ABB IRB 120) are represented using Denavit–Hartenberg (DH) parameters. To simulate the robot, the robotics toolbox developed by Peter Corke was used [10].

### *2.3.1 Robot Inverse Kinematics*

After generating the Cartesian trajectories for the print, the robot tooltip should track the waypoints in each trajectory. Therefore, it is necessary to determine the joint configurations for each of these end-effector poses along the trajectories. This process is called inverse kinematics (IK). The solution of the IK problem can be obtained analytically or numerically.



(a) (b)

Fig. 8: a) Trajectories in a single cylinder b) Trajectories in multiple cylinders

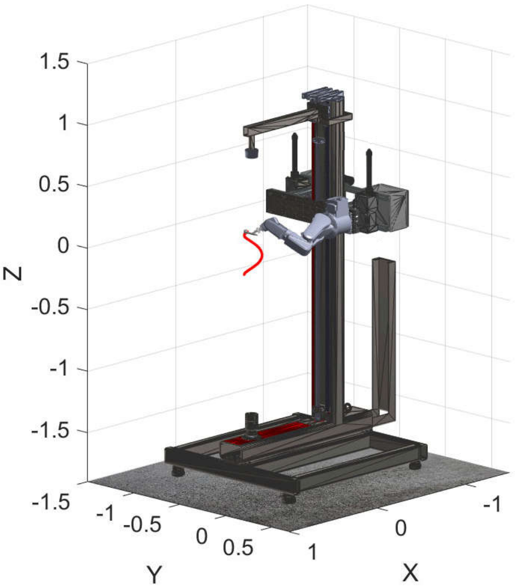


Fig. 9: Simulation environment and robot model

In the analytic method, a model of the robot is created, which defines the relationship between joints and, algebra is used to solve for joint variables. Analytic solutions become increasingly difficult to solve when the number of joints increases [31]. Especially when there is a high degree of freedom, obtaining an analytical solution is very difficult or sometimes impossible[13]. The robot used in this research has 6 degrees of freedom thus it is difficult to find a solution analytically [22]. Therefore, a numerical solution has been used in this application. In the numerical solution, firstly, forward kinematics (FK) is used to determine the end-effector location. FK in robotics maps each joint configuration to an end-effector Cartesian pose (i.e. a 3D point and orientation) [28]. We can determine the end-effector pose using FK, $K(q)$ and since we know the desired pose, $\xi^*$ which is a waypoint in the printing trajectory, then the general problem (21) can be framed as an optimisation problem whereby the error between the two aforementioned poses is minimised as shown in (22).

$$q = K^{-1}(\xi^*) \tag{21}$$

$$q^* = \underset{q}{\text{argmin}} \|K(q) \ominus \xi^*\| \tag{22}$$

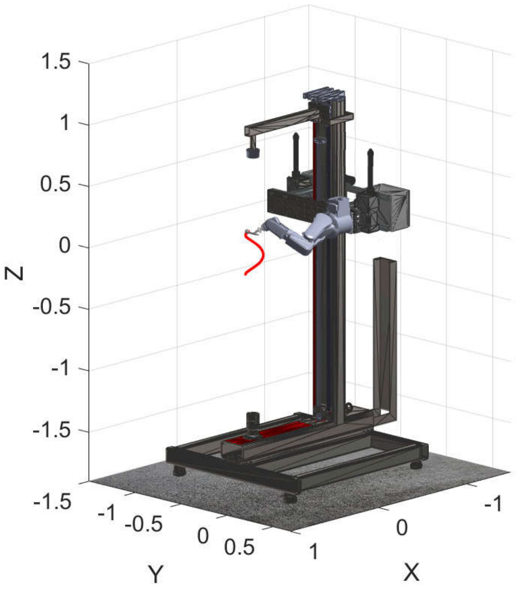


Fig. 10: Printing a trajectory at the beginning of the print

*2.3.2 Manipulability*

The dexterity of a robotic manipulator or its ability to move easily in any arbitrary direction is referred to as its manipulability [11]. This is a scalar measure where a higher value is better. This value can be calculated for each point along a particular tool path. This paper evaluates this measure to use with 3D printers to compute how easy it is to execute a tool path to print an object in a given location for any given industrial robot manipulator. Manipulability is based on the Jacobian matrix of the manipulator. Jacobian matrix maps joint rates, $\dot{q}$ to end-effector Cartesian velocity, $v$ as shown in (23), where $q$ is the joint angles [11]. In this research, Yoshikawa's manipulability measure [38] is been used as shown in (24).

$$\dot{q} = J(q)^{-1}v \tag{23}$$

$$m = \sqrt{det\left(JJ^{T}\right)} \tag{24}$$

## 3 Results

### 3.1 Output of the Slicing Algorithm

To simulate the realistic printing process, a set of fine-grained parameters have been selected. The parameter values are $\Delta r = 5\ mm$, $\Delta C = 5\ mm$, $\Delta h = 2\ mm$, $r_{end} = 65\ mm$, $r_{start} = 330\ mm$ and the total height of the spiral is approximately 1.3 $m$. After the slicing, all points were exported as a Polygon File Format (PLY) file for visualisation, and trajectories are saved as a Matlab data file for simulation purposes. The final sliced output contains around 250000 vertices. Waypoints in the printing trajectories are saved in a PLY file. Figure 13a shows waypoints in the printing trajectories while Fig. 13b shows these points overlapped with the STL model of the actual spiral. This overlapping shows that the generated trajectories are in the shape of the original object. To test the presented method, several helical shapes (other than a GSS) have been sliced: a uniform solid coil representing a suspension spring used in large scale machinery (shown in Fig. 12a); and an arbitrary wavy cross-section (shown in Fig. 12c). The slicing results are shown in Fig. 12b and Fig. 12d. Figure 12a shows the STL model overlapped with intersection points which provides an indication of an accurate shape approximation and Fig. 12b shows the generated trajectories for the robot arm to follow. Since a plot of all trajectories would result in the lines being too close for visualisation, only every 5th slicing cylinder and every 5th trajectory in a given cylinder (i.e. 4% of all trajectories) is shown. These generated trajectories are also true to the original shape that was sliced. Figure 12c shows another helical shape with an arbitrary cross-section which is quite different from a GSS. Figure 11 shows a closer view of the intersection points created during the slicing process, and the sampled trajectories are shown in Fig. 12d.

The GSS model was sliced using the conventional Z-direction planar slicing to demonstrate the difference between the two slicing methods. Due to print direction, a large number of support structures are required as shown in Fig. 14. The green structures indicate support material, while the brown demonstrates the spiral which is required to be printed. Therefore, if a Z-directional slicing and printing method is followed, all those printed green colour support structures need to

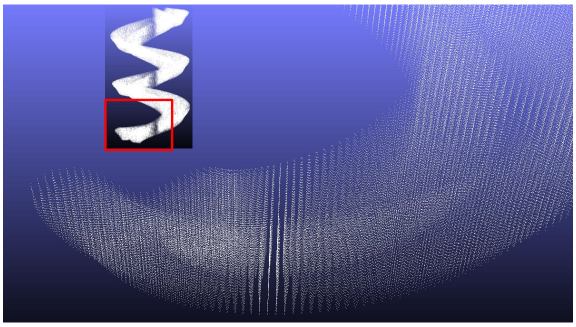
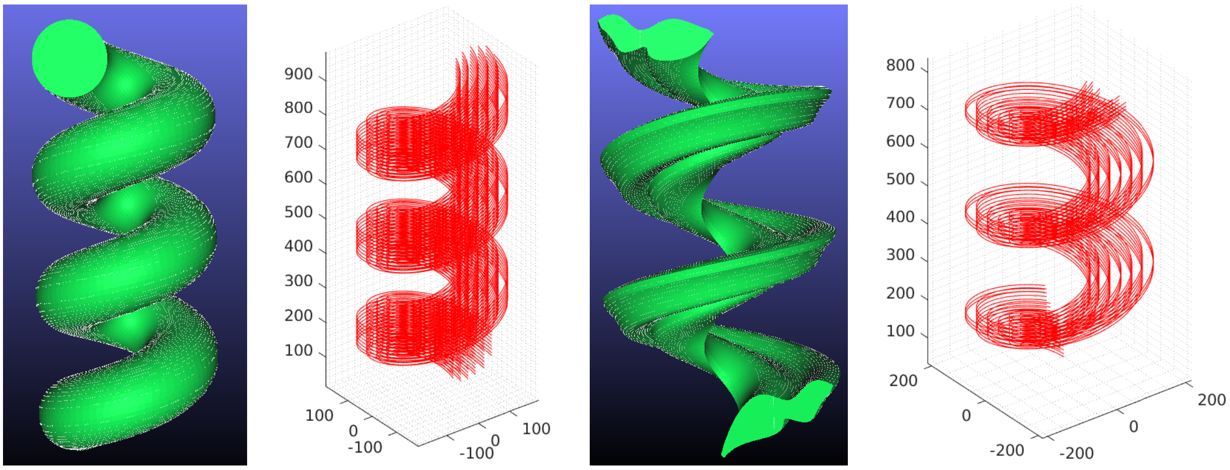


Fig. 11: Closer view of the intersection points at the bottom



(a) (b) (c) (d)

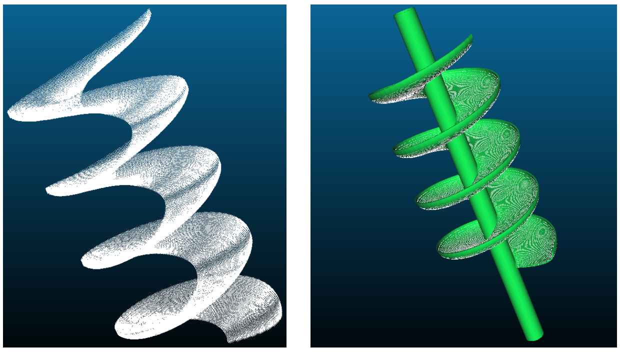
Fig. 12: a) Solid coil (green) overlapped with intersection points (white), b) Generated trajectories showing only every 5th slicing cylinder and every 5th trajectory in a cylinder (i.e. 4% of all trajectories), c) Helical spiral with an arbitrary cross-section (green) overlapped with intersection points (white), d) Generated trajectories showing only every 5th slicing cylinder and every 5th trajectory in a cylinder (4% of all trajectories)

be removed to obtain the final shape. Printing this support material (green structures) alone takes a lot of time compared to printing the required shape (brown areas). Removing support structures is a tedious process after printing and it can reduce the surface smoothness, which will reduce the performance of the spiral. Additionally, printing support structures increases the cost of printing due to the additional material usage. Radial slicing and printing removes this problem because the printing process occurs perpendicular to the Z-axis.

### 3.2 Printed Spirals with Manipulability Measure

Simulation experiments of the printing according to the generated tool paths were conducted in the previously mentioned simulation environment. Figure 10 shows the beginning of the simulation and the printing of the first helical trajectory. For each location that the robot arm moves during printing, the manipulability measure was calculated as detailed in 2.3.2. The location of the end effector and the manipulability values are saved to a file and analysed using the Cloud Compare software. This allows potential print faults to be identified and the manipulability measure to be visualised in various locations of the print as shown in Fig. 15.

The future objective of this research is to identify ideal locations to print sensors in line with the spiral



(a) (b)

Fig. 13: a) Exported PLY file from slicing algorithm b) Exported PLY file overlapped with original STL file

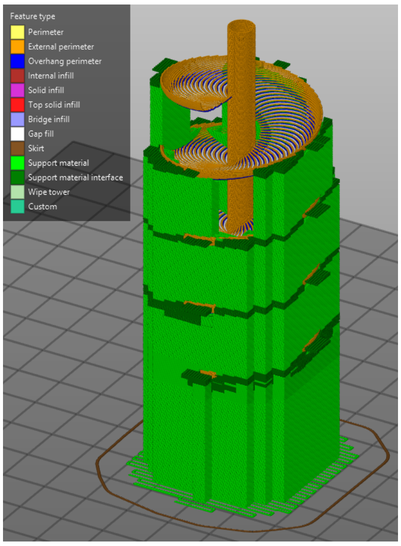
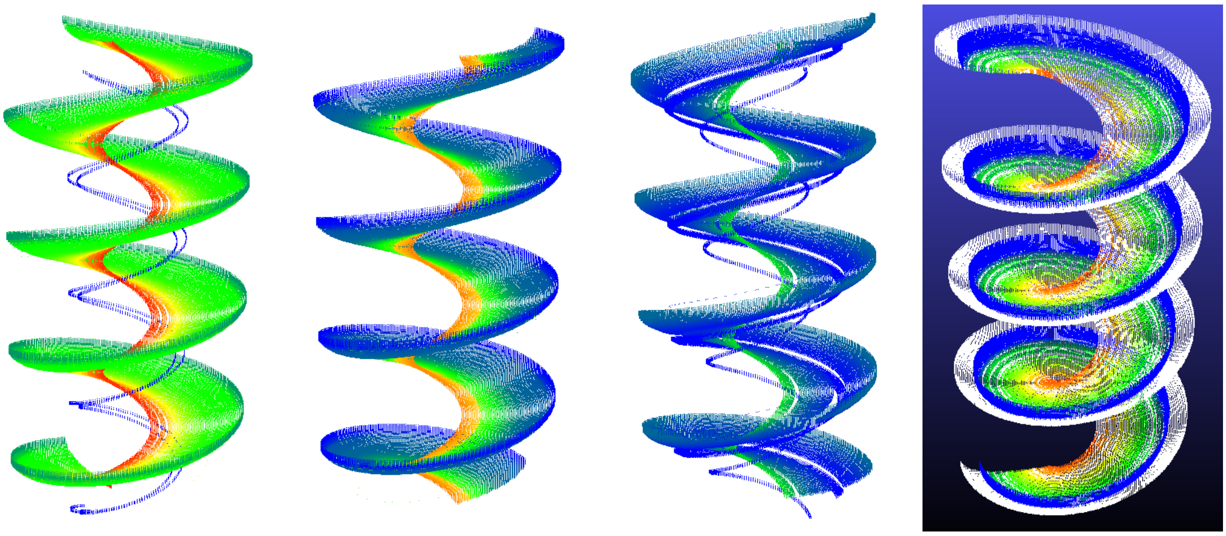


Fig. 14: Result from conventional Z-direction slicing



(a) (b) (c) (d)

Fig. 15: Examples of successful and failed prints. a) Failed print - case 1 b) Successful print - case 2 c) Failed 90° angle print d) Failed 0° angle print (coloured points) overlapped with trajectory waypoints (white colour points) showing a shrunken failed print. The colours represent the amount of manipulability where red, yellow, green and blue shows the transition from high to low manipulability values

print. A map of manipulability enables the system to better plan arm motions that improve dexterity and, in turn, increases the likely hood that 3D printed sensors can be easily and accurately incorporated. There are a high number of locations and poses a robot arm can have when executing a printing tool path. Depending on the desired tool orientation, the output of the print might need to vary slightly (or significantly) because of the limited amount of manipulability of the robot. To print each cylinder, transitioning from one cylinder to the next cylindrical layer (i.e. one "layer height" radius increase) can be done in different ways. Figure 16 shows two such cases. In case 1, the print head moves from the innermost cylinder to the outermost cylinder in a 45° angle. In case 2, a few inner cylinders (indexes 1-11) were printed in the $-X$ direction. Then, to print the rest of the cylinders, the print head was rotated 45° as shown in Fig. 15b.

Figure 15 includes the results of these two cases. Figure 15a shows case 1 results and Fig. 15b shows case 2 results. The colours in the aforementioned figures represent the manipulability values as shown in the histograms in Fig. 17 and Fig. 18. According to the case 1 results shown in Fig. 15a, the robotic arm was not

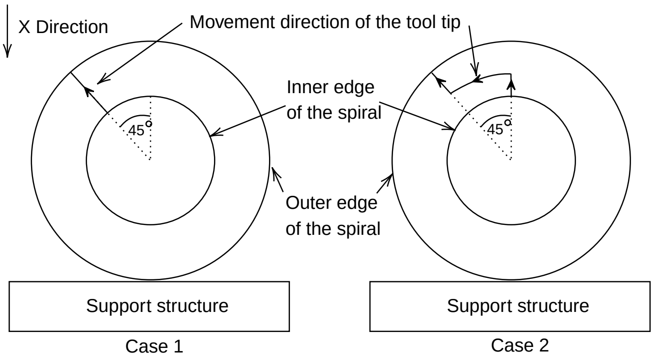


Fig. 16: Two examples of print head movements

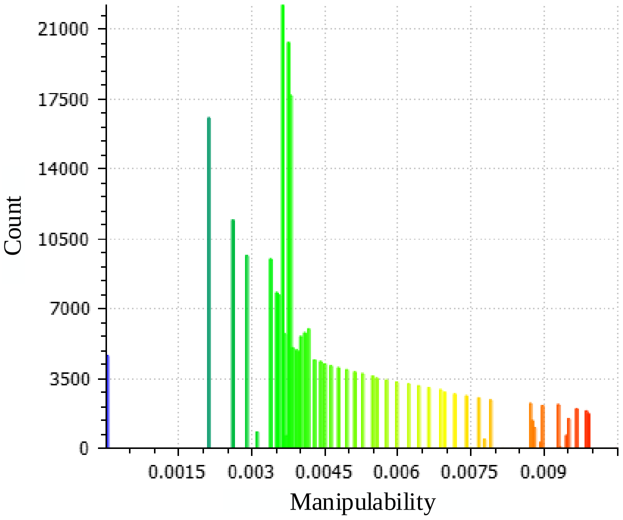


Fig. 17: Manipulability histogram of failed print

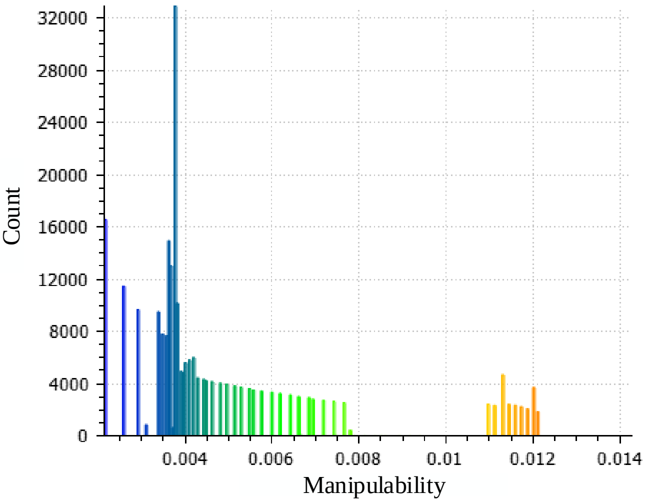


Fig. 18: Manipulability histogram of successful print

able to execute 3% of the 52 tool paths in the middle, because of an inability to find joint trajectories for the robot that would place the end effector at the required points along the tool path. There were several failed print scenarios for different cases as shown in Fig. 15c and Fig. 15d. Figure 15d shows a shrunken print where white dots are waypoints of the trajectories a printer should follow, and the coloured points are the printed spiral. This print failed because the robot arm was not able to move to the furthest layers. Conversely, Fig. 15b shows a successful result from case 2, where the overlap is seen to be correct. Case 2 was identified by heuristic analysis of observations from other prints with similar shapes. When the manipulability value is near zero (indicated by the blue colour in the histogram), the ability to print with the robotic arm or the dexterity of the arm is low. Therefore, this simulation displaying the manipulability measure is a good indication of the predicted print output (i.e. how likely it is to succeed), as well as the dexterity of the manipulator as it moves through the tool path. This is particularly interesting for future research where sensor placement will need to be simultaneously considered in conjunction with the printing process.

To understand the relationship between successful prints and robot manipulability, radial manipulability values have been plotted for all test cases in Fig. 15 and in Fig. 19. For case (a), there are two sudden drops in manipulability along the radius, which correlate with the two failed areas in Fig. 15a. For the successful print case (Fig. 15b), there was no sudden change in manipulability except during a change in the print direction as explained earlier and shown as case 2 in Fig. 16. Similar to case (a), there were two sudden drops in manipulability in case (c). However, the severity of the second drop in manipulability (down to near zero) effectively removes the robot arm's capability to move in a single degree of freedom. For the case (d), from Fig. 19, it can be seen that the manipulability reached zero towards the outer cylindrical layers, indicating the robot lost a degree of freedom, and resulting in the misshapen print as shown in Fig. 15d.

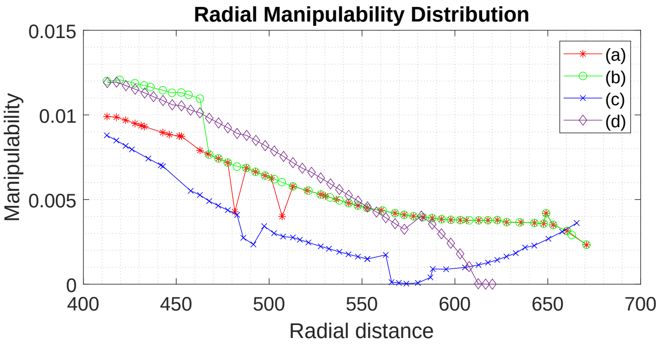


Fig. 19: Radial manipulability distribution in different spiral prints

When considering manipulability as a printability measure, the likely success of the print should be identified using the local manipulability measure rather than the overall average manipulability during the trajectory. While the robot arm may exhibit high manipulability throughout the majority of the trajectory, any situation where the robot arm reaches a near zero manipulability will effectively cause the print to fail. This is due to the robot arm's inability to move in one or more directions (X, Y or Z), or orientations (roll, pitch or yaw), from arm configurations with a near zero manipulability measure. Case (d) in Fig. 19 highlights this effect with the arm becoming unable to move outwards, resulting in a shrunken and misshapen print. To show this more clearly, Table 1 includes the average cross-section of manipulability during each print. Case (b) is the successful print with the second-highest average manipulability. Despite case (d) having a higher average manipulability measure than case (b), the occurrence of a near-singular configurations towards the end of the print trajectory caused its failure, reinforcing the

Table 1: Average cross-section manipulability in spiral prints

| Case | Average Manipulability |
|---|---|
| (a) | 0.0057 |
| (b) | 0.0061 |
| (c) | 0.0031 |
| (d) | 0.0068 |

need to use the localised rather than a global average measures of manipulability. This is advantageous since the manipulability of a robot arm can be calculated in real-time during printing. Furthermore, if a failure is detected, the print can be halted, thus reducing material waste from an incorrect print job, especially one as time consuming as a large scale GSS print.

## 4 Discussion

Most of the widely used slicing algorithms in current literature perform slicing in the Z direction [1,27,12]. These methods are not suitable for large helical objects due to the excessive amount of support material required (as shown in Fig. 14). The undesirable support material, results in a significant waste of resources and the potential to improperly roughen or even damage the quality of the printed GSS surface, when such such support material is removed. Some researchers have proposed curved slicing algorithms [40,21,6] however they they are unsuitable for this application due to constraints such as special manufacturing methods (i.e. LOM), or the limitation of only creating thin, small-scale parts. Therefore, the authors developed a novel radial slicing algorithm to overcome these limitations and print large helical objects around a rotating column.

The slicing algorithm generates tool paths to radially slice the helical spiral shape, and the simulations enable the tool paths to be visualised and analysed. This slicing algorithm is developed for a bespoke printer to manufacture industrial equipment in a helical shape with a solid fill. However, this slicing algorithm is not designed to handle hollow structures since infills are required to print overhanging areas. The authors demonstrated the relationship between local manipulability and print failures. For large scale prints, real-time failure identification will reduce resource and time wastage by terminating inaccurate prints. The fact that different radial start points can be seen to affect the manipulability of the robot, demonstrates the need to investigate path planning methods which ensure optimal robotic arm manipulatively during the print. The authors are conducting further research to determine the optimal tooltip path based on an optimisation of a weighted cost function of manipulability, location error and rotation error. Furthermore, the integration of additional tasks that need to be executed by the robot whilst printing, such as sensor placement, highlights the need to investigate the inclusion of seamless movements with the printing tool paths that are generated by the slicer.

## 5 Conclusion

This paper has presented a novel slicing algorithm that can be used to slice a large scale helical object, such as a spiral, which must be printed in a radial manner from the centre to outwards, as opposed to layers on Z-direction during conventional AM printing. Different helical shapes have been sliced and simulated, showing that the tool paths generated by the presented slicing algorithm can be used to print the desired shape with an industrial robot. The authors have shown that the presented slicing method will overcome the limitations of Z-direction slicing, such as undesirable step-wise surface roughness and the wasteful printing of support structures. The insights into robot manipulability has been explored to enable the prediction of successful prints, and to demonstrate the effect of dexterity. This synergy of tool path generation, alongside an understanding of the robot capability, opens up new research possibilities into the incorporation of the simultaneous object and sensor placement while printing. Future research will determine optimal printing tooltip paths based on an optimisation of a weighted cost function of manipulability, location error and rotation error, and will culminate in conducting real-world experiments on the full-scale printer.

## 6 Acknowledgement

This research is supported by UTS, The Commonwealth of Australia's Department of Industry, Innovation and Science (Innovative Manufacturing CRC Ltd) and Downer, via its subsidiary Mineral Technologies. Thank you to Rapido, in particular, Hervé Harvard and Michael Behrens for establishing this overall research activity and leading the overall R&D engineering project. Authors also thank Jordan Henry for creating the simulation environment of the printer according to the actual dimensions. Additionally, we also thanks Sheila Sutjiptu and Yujun Lai for providing robotics-related feedback. We also thank UTS:RI /CAS for providing required resources to carry out this research.

## References

1. Adnan, F.A., Romlay, F.R.M., Shafiq, M.: Real-time slicing algorithm for Stereolithography (STL) CAD model applied in additive manufacturing industry. IOP Conference Series: Materials Science and Engineering **342**(1), 012016 (2018)
2. Australia, E.: Milestone for mining manufacture with 3D printing. URL `https://portal.engineersaustralia.org.au/news/milestone-mining-manufacture-3d-printing`

3. Borst, C., Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure : Representing robot capabilities Capturing Robot Workspace Structure : Representing Robot Capabilities. December 2007, pp. 3229–3236 (2016)
4. Camacho, D.D., Clayton, P., O'Brien, W., Ferron, R., Juenger, M., Salamone, S., Seepersad, C.: Applications of Additive Manufacturing in the Construction Industry. In: ISARC, vol. 34, pp. 1–8. Vilnius (2017)
5. Cao, W., Miyamoto, Y.: Direct Slicing from AutoCAD Solid Models for Rapid Prototyping. International Journal of Advanced Manufacturing Technology **21**(10-11), 739–742 (2003)
6. Chakraborty, D., Aneesh Reddy, B., Roy Choudhury, A.: Extruder path generation for Curved Layer Fused Deposition Modeling. Computer-Aided Design **40**(2), 235–243 (2008)
7. Chang, C.C.: Direct slicing and G-code contour for rapid prototyping machine of UV resin spray using PowerSOLUTION macro commands. International Journal of Advanced Manufacturing Technology **23**(5-6), 358–365 (2004)
8. Cheng, G.Z., Folch, E., Wilson, A., Brik, R., Garcia, N., Estepar, R.S.J., Onieva, J.O., Gangadharan, S., Majid, A.: 3D Printing and Personalized Airway Stents. Pulmonary Therapy **3**(1), 59–66 (2017)
9. Choi, S.H., Kwok, K.T.: Hierarchical slice contours for layered-manufacturing. Computers in Industry **48**(3), 219–239 (2002)
10. Corke, P.: Guestbook: Robotics Toolbox. URL `http://www.petercorke.com/RTB/`
11. Corke, P.: Robotics Vision and Control, vol. 75, 2 edition edn. Springer (2015)
12. Feng, J., Fu, J., Lin, Z., Shang, C., Li, B.: Direct slicing of T-spline surfaces for additive manufacturing. Rapid Prototyping Journal **24**(4), 00–00 (2018)
13. Gan, J.Q., Oyama, E., Resales, E.M., Hu, H.: A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm. Robotica **23**(1), 123–129 (2005)
14. Gök, A., Gök, K., Bilgin, M.B., Alkan, M.A.: Effects of cutting parameters and tool-path strategies on tool acceleration in ball-end milling. Materiali in tehnologije **51**(6), 957–965 (2017)
15. Gok, A., Gologlu, C., Demirci, H.I.: Cutting parameter and tool path style effects on cutting force and tool deflection in machining of convex and concave inclined surfaces. The International Journal of Advanced Manufacturing Technology **69**(5-8), 1063–1078 (2013)
16. Haipeng, P., Tianrui, Z.: Generation and optimization of slice profile data in rapid prototyping and manufacturing. Journal of Materials Processing Technology **187-188**, 623–626 (2007)
17. ISO;ASTM: Standard Terminology for Additive Manufacturing – General Principles – Terminology. Tech. rep. (2015)
18. Jing Hu: Study On STL-Based Slicing Process For 3D Printing. In: Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium, pp. 885–895 (2017)
19. Kim, H.J., Wie, K.H., Ahn, S.H., Choo, H.S., Jun, C.S.: Slicing algorithm for polyhedral models based on vertex shifting. International Journal of Precision Engineering and Manufacturing **11**(5), 803–807 (2010)
20. Kirschman, C.F., Jara-Almonte, C.C.: A parallel slicing algorithm for solid freeform fabrication processes. In: International Solid Freeform Fabrication Symposium, p. 26–33 (1992)
21. Klosterman, D.A., Chartoff, R.P., Osborne, N.R., Graves, G.A., Lightman, A., Han, G., Bezeredi, A., Rodrigues, S.: Development of a curved layer LOM process for monolithic ceramics and ceramic matrix composites. Rapid Prototyping Journal **5**(2), 61–71 (1999)
22. Manocha, D., Canny, J.F.: Efficient inverse kinematics for general 6R manipulators. IEEE Transactions on Robotics and Automation **10**(5), 648–657 (1994)
23. Munasinghe, M.I.N.P., Miles, L., Paul, G.: Direct-Write Fabrication of Wear Profiling IoT Sensor for 3D Printed Industrial Equipment. In: ISARC, pp. 862–869 (2019)
24. Munasinghe, N., Paul, G.: Advanced Manufacturing of Spirals for Mineral Separation with Integrated Smart Sensing. In: IEEE UNITE (2019). URL `http://hdl.handle.net/10453/135218`
25. Munasinghe, N., Paul, G.: Ultrasonic-Based Sensor Fusion Approach to Measure Flow Rate in Partially Filled Pipes. IEEE Sensors Journal **20**(11), 6083–6090 (2020)
26. Munasinghe, N., Woods, M., Miles, L., Paul, G.: 3-D Printed Strain Sensor for Structural Health Monitoring. In: Int. Conf. CIS-RAM. IEEE, Bangkok (2019)
27. Oropallo, W., Piegl, L.A., Rosen, P., Rajab, K.: Point cloud slicing for 3-D printing. Computer-Aided Design and Applications **15**(1), 90–97 (2018)
28. Parikh, P.J., Lam, S.S.Y.: A hybrid strategy to solve the forward kinematics problem in parallel manipulators. IEEE Transactions on Robotics **21**(1), 18–25 (2005)
29. Porges, O., Lampariello, R., Artigas, J., Wedler, A., Grunwald, G., Borst, C., Roa, A.: Reachability and Dexterity: Analysis and Applications for Space Robotics. Proceedings of 13th Symposium on Advanced Space Technologies in Robotics and Automation-ASTRA 2015 (3), 1–7 (2015)
30. Shakor, P., Nejadi, S., Paul, G., Malek, S.: Review of Emerging Additive Manufacturing Technologies in 3D Printing of Cementitious Materials in the Construction Industry. Frontiers in Built Environment **4**(January) (2019)
31. Srisuk, P., Sento, A., Kitjaidure, Y.: Inverse kinematics solution using neural networks from forward kinematics equations. In: 2017 9th International Conference on Knowledge and Smart Technology (KST), pp. 61–65. IEEE (2017)
32. Sturges, R.H.: A Quantification of Machine Dexterity Applied to an Assembly Task. The International Journal of Robotics Research **9**(3), 49–62 (1990)
33. Sun, S.H., Chiang, H.W., Lee, M.I.: Adaptive direct slicing of a commercial CAD model for use in rapid prototyping. The International Journal of Advanced Manufacturing Technology **34**(7-8), 689–701 (2007)
34. Sutjipto, S., Tish, D., Paul, G., Vidal-Calleja, T., Schork, T.: Towards Visual Feedback Loops for Robot-Controlled Additive Manufacturing. In: Robotic Fabrication in Architecture, Art and Design 2018, pp. 85–97. Springer International Publishing (2019)
35. Szilvási-Nagy, M., Mátyási, G.: Analysis of STL Files. Mathematical and Computer Modelling **38**(7-9), 945–960 (2003)
36. Wong, K.V., Hernandez, A.: A Review of Additive Manufacturing. ISRN Mechanical Engineering **2012**(208760), 10 (2012)
37. Yang, P., Oian, X.: Adaptive slicing of moving least squares surfaces: Toward direct manufacturing of point set surfaces. Journal of Computing and Information Science in Engineering **8**(3), 310031–3100311 (2008)

38. Yoshikawa, T.: Manipulability and redundancy control of robotic mechanisms. In: Proceedings. 1985 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1004–1009. Institute of Electrical and Electronics Engineers (1985)
39. Zhang, Z., Joshi, S.: An improved slicing algorithm with efficient contour construction using STL files. International Journal of Advanced Manufacturing Technology **80**(5-8), 1347–1362 (2015)
40. Zhao, G., Ma, G., Feng, J., Xiao, W.: Nonplanar slicing and path generation methods for robotic additive manufacturing. International Journal of Advanced Manufacturing Technology **96**(9-12), 3149–3159 (2018)
41. Zhou, M.Y.: STEP-based approach for direct slicing of CAD models for layered manufacturing. International Journal of Production Research **43**(15), 3273–3285 (2005)