



Article A Robot-Centered Path-Planning Algorithm for Multidirectional Additive Manufacturing for WAAM Processes and Pure Object Manipulation

Markus Schmitz ^{1,*}, Jan Wiartalla ¹, Markus Gelfgren ¹, Samuel Mann ², Burkhard Corves ¹, and Mathias Hüsing ¹

- ¹ Institute of Mechanism Theory, Machine Dynamics and Robotics (IGMR), RWTH Aachen University, Eilfschornsteinstraße 18, 52064 Aachen, Germany; wiartalla@igmr.rwth-aachen.de (J.W.); markus.gelfgren@rwth-aachen.de (M.G.); corves@igmr.rwth-aachen.de (B.C.); huesing@igmr.rwth-aachen.de (M.H.)
- ² Welding and Joining Institute (ISF), RWTH Aachen University, Pontstraße 49, 52062 Aachen, Germany; mann@isf.rwth-aachen.de
- * Correspondence: schmitzm@igmr.rwth-aachen.de; Tel.: +49-241-80-99801

Abstract: Previous algorithms for slicing, path planning or trajectory planning of additive manufacturing cannot be used consistently for multidirectional additive manufacturing with pure object manipulation in wire-arc additive manufacturing. This work presents a novel path planning approach that directly takes robot kinematics into account and thus ensures the reachability of all critical path poses. In an additional step, the planned path segments are smoothed so that joint velocity limits are respected. It is shown that the implemented path planner generates executable robot paths and at the same time maintains the process quality (in this case, sufficient coverage of the slice area). While the introduced method enables the generation of reachable printing paths, the smoothing algorithm allows for the execution of the path with respect to the robot's velocity limits and at the same time improves the slice coverage. Future experiments will show the realization of the real robot setup presented.

Keywords: multidirectional additive manufacturing; WAAM; additive manufacturing; path planning

1. Introduction

Multidirectional additive manufacturing provides promising opportunities ranging from the freedom of design of components to the targeted influencing of component properties compared to traditional 2.5D additive manufacturing methods. For wire-arc additive manufacturing, especially with eccentric wire feeding or parallel sensor technology, there is great potential for process realization. The permanent fixation of the welding head in the workspace of the robot and the pure manipulation of the component by the robot in relation to the welding head is characteristic. The six-dimensional requirements for preprocessing (slicer, path planner and trajectory planner) cannot be ensured by algorithms and methods used for additive manufacturing or robotic coverage path planning so far. Traditionally, the kinematics and limits of the machine or robot executing the print are only considered during the last pre-processing step of trajectory generation before execution. In multidirectinal additive manufacturing, this procedure usually results in paths that are not executable due to unreachable path poses or high joint velocities. A major deficit comes from the lack of consideration of the robot's kinematics as well as its limits already during path planning. This deficit is counteracted in this work by introducing a novel robot-centered path planning method. The method is based on the decomposition of each slice into multiple convex polygons, subsequent evaluation of possible infill strategies for each polygon and selection of the most suitable infill combination by a modified Hamilton



Citation: Schmitz, M.; Wiartalla, J.; Gelfgren, M.; Mann, S.; Corves, B.; Hüsing, M. A Robot-Centered Path-Planning Algorithm for Multidirectional Additive Manufacturing for WAAM Processes and Pure Object Manipulation. *Appl. Sci.* 2021, *11*, 5759. https://doi.org/ 10.3390/app11135759

Academic Editors: Namhun Kim, Seung Ki Moon and Rohan Shirwaiker

Received: 19 May 2021 Accepted: 16 June 2021 Published: 22 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Graph search. It is shown that the methodology is capable of generating complex sixdimensional paths, which are executable for a specified robot. This includes both the pure reachability of path poses as well as the adherence of joint velocity limits through the smoothing of directional changes in the path. Effects on the process can be taken into account in each process step.

2. Multidirectional Additive Manufacturing

2.1. General Concept

Multidirectionality in the context of additive manufacturing (AM) refers to the degrees of freedom of the process. However, the decisive factor is the possibility to build up layers in all directions or even to detach from the flat layered structure (Figure 1). In contrast to traditional AM processes, where a limited adjustment of the orientation between print bed and print head is possible, multidirectionality explicitly requires a significant variation of the build-up direction compared to the initial direction. It is generally not relevant whether the print head or print bed is moved.



Figure 1. Illustration of multidirectionality with variable building direction of: (**a**) subvolumes; and (**b**) curved surfaces.

The term *object manipulation* is used to create a clear distinction from related manufacturing processes of multidirectional additive manufacturing. Object manipulation thus describes the additive construction of a component by exclusively manipulating the object relative to a fixed print head. This work focuses on multidirectional additive manufacturing by means of object manipulation. Figure 2a illustrates the advantages of pure object manipulation, which leads to the exclusive production in a downhand welding position due to the fixed print head. For Wire-Arc Additive Manufacturing (WAAM), multi-wire applications have the special potential to further increase productivity or enable multi-material applications for tailor-made material properties that are difficult to realize with common powder bed fusion processes [1].

For the process with eccentric wire feeding (or alternatively stationary sensor technology or other weld head periphery), Figure 2b illustrates that manipulating a weld head along a path, the entire weld head periphery must always be rotated together with the weld head. As a result, complex paths can often not be executed. For this application, pure object manipulation is therefore advantageous. If multidirectional additive manufacturing by means of object manipulation (MDAM) is to be extended to the application of WAAM with eccentric wire feeding, this is accompanied by the loss of the degree of freedom around the printing axis of the print or weld head in 5D printing [2]. This has the following reasons:

- A wire feed must mainly be fed from the front, or at least from a defined direction in relation to the direction of movement, because the geometry of the deposited material is highly dependent on the relative alignment of the wire.
- Sensors used for monitoring and controlling the process should continuously analyze the weld seam and must therefore always be aligned with the welding process.



Figure 2. Object-manipulation showing advantages through: (**a**) downhand welding position; and (**b**) fixing of the periphery.

2.2. Process Chain

The process chain of MDAM is closely modeled on known AM process chains. In contrast to the processes of conventional 3D printing, the manipulator used for MDAM plays a central role (Figure 3). On the one hand, the early definition of a robot (before starting the pre-processing) ensures the feasibility of the process for a specific robot and, on the other hand, opens up scope for optimization along the process chain.





Starting from the workpiece data in STL format, the workpiece is divided into subvolumes and then slices. Ding et al. introduced the dividing algorithms used in this work [3]. The slicing algorithm directly takes multidirectionality into account. Printing paths can then be planned in each of the generated slices, which in turn are transferred into an executable robot trajectory. After physical production, components of the WAAM process often have to undergo post-processing. The availability of path and trajectory planning can in turn be used for process control interventions. On the one hand, essential information about the target geometry can be derived from path planning. On the other hand, information about expected deviations can be passed on from path and trajectory planning to the process control system to ultimately correct the workpiece quality via the welding process parameterization.

2.3. Major Challenges

The mentioned high demands on the manufacturing process caused by the use of industrial robots motivate the parallel consideration of the applied robot (and periphery) already from the beginning of the planning process [4]. Existing algorithms of slicing and path planning from conventional AM are not executable for the robot [5]. The executability can essentially be attributed to the pure reachability of the robot's poses and the manipulability of the robot at the required high printing velocities. Once the path has been planned, it can only be influenced by deviations from the predefined path within the scope of trajectory planning. This contribution shows that poses which can be reached in this way can be brought below the speed limits by slight deviations from the path to be printed. The remaining unreachable path segments can only be adjusted by modifying the path planning. This motivates the development of a path planner that takes into account both the reachability and the manipulability of the manipulator and at the same time has a minimal negative impact on other process parameters and printing result quality [6].

3. Path Planning in Robotics and Additive Manufacturing

3.1. Path Planning in Additive Manufacturing

In the field of path planning, it is necessary to distinguish between the different possibilities of degrees of freedom of movement. Paths can be planned either freely in three-dimensional space or two-dimensionally within a plane. In AM, the printing process must ensure the positioning and orientation of the print head in relation to the print bed. In a conventional fused deposition modeling (FDM) process, it is sufficient to position the print nozzle at the designated position and always keep it perpendicular to the print bed. In contrast, the orientation of the print head in all three rotational degrees of freedom is of great importance for the present work. Changing the direction of the external wire feed to the path tangent would have a decisive influence on the quality of the welded path.

Despite the common designation of AM as '3D printing', an actual execution of paths in space is rarely found, as most processes can be strictly classified as 2.5D. The path planning of AM processes has already been researched and developed in many ways. The focus of path planning is on generating a component with the highest possible accuracy, high surface quality and good mechanical properties. Various infill strategies have been developed for this purpose. Typical strategies are raster, zigzag, contour, spiral, spacefilling, grid, honeycomb, hexagonal or Voronoi diagram-based infills (cf. Figure 4). The different strategies are not always suitable for use in WAAM-based processes. In addition, certain space-filling infills are not executable by a robot due to movement limits or require many changes of direction. Therefore, in the context of this work, the simplest infills (namely raster and zigzag) are considered first. Table 1 gives an overview of different path planning algorithms. The summary describes the advantages and disadvantages in the context of robotic MDAM using WAAM.

It can be summarized that none of the presented approaches, methods or algorithms can be directly applied to MDAM, especially for pure object manipulation in WAAM. The reason for this is the lack of integration of robotic parameters in the path planning of AM.



Figure 4. Common infill strategies in AM.

Table 1. Summary of the state of the art of path planning algorithms. Advantages and disadvantages are summarized in thecontext of robotic MDAM using WAAM.

Reference	Relevant Developments	Advantages	Disadvantages	
Zhang et al. [7]	change of building direction and curves surfaces/slices (cf. Figure 5a)	necessary subvolumization approach	restricted range for building direction	
Fang et al. [8]	segmented consideration of paths (cf. Figure 5b)	include process parameter into optimization, optimize robot motion	use of rotational degree of freedom around the welding head	
Simplify3D [9] (cf. Figure 5c)	subdivision of objects to be printed into an outer shell and an infill structure	-	2.5D Slicer without any optimization	
Dunlavey et al. [10]	raster infill strategy	easy process handling	2.5D Slicer	
Park et al. [11]	zigzag infill strategy	continuous path, reduced number of sub-paths, fewer transition movements	many changes in printing direction	
Huang [12]	dividing slices into many convex polygons, optimization algorithm generates optimal zigzag infill orientation	minimal number of changes in direction	-	
Farouki et al. [13]	contour infill strategy (Figure 4)	reduce mistakes near the contour	orientation changes more then 180 deg	
Wang et al. [14]	spiral infill strategy	used in 5D numerically controlled (NC) machining, solve the limitations of zigzag infill	only suitable for special geometric model, orientation changes more then 180 deg	
Zhang et al. [15]	hybrid infill structure, zigzag+contour	combine the advantages of several strategies	orientation changes more then 180 deg	
Jin et al. [16]	hybrid infill structure, zigzag+contour	combine the advantages of several strategies	orientation changes more than 180 deg	
Bertoldi et al. [17]	Hilbert infill structure	cover complex polygons, reducing shrinkage in AM processes	large number of path orientation changes, not suitable for WAAM processes [18]	
Wasser et al. [19]	fractal-like path planning using a simulated annealing algorithm, area to be deposited is broken down into nodes	possible path optimization	for large areas and high accuracy requirements, the processing time required is unacceptably long	
Dwivedi and Kovacevic [20]	hybrid infill structure, zigzag + continuous path strategies, decomposed into a set of monotone polygons	reduces the number of welding paths	-	
Ding et al. [21]	hybrid infill structure, zigzag + contour + continuous path patterns	continuous path pattern is suitable for WAAM of solid structures	orientation changes more than 180 deg	
Lin et al. [22]	maze-like structure, decompose slice and map volumes in weighted graph, find optimal path by backtracking algorithm	increase the isotropy of the mechanical properties in the component	many orientation changes, not suitable for WAAM processes [18]	
Schmitz et al. [6]	Hamilton graph search, decompose slice and map pixels in weighted graph, find optimal path using a backtracking algorithm	evaluate robot motion along printing path	many orientation changes, calculation effort not suitable, hard to find a single Hamilton path	



Figure 5. Example applications of path planning in AM: (**a**) AM of a hollow wing structure [7]; (**b**) path optimization with regard to heat input during welding [8]; and (**c**) slicing software Simplify3D with visible outer contour and filling structure [23].

3.2. Path Planning in Robotics

In general, trajectory planning can be understood as the determination of motion profiles for the drive system of an automated machine and thus represents the link between space and time. Knowledge of the specific kinematics (forward and inverse kinematics) is indispensable for this purpose. In order to be able to describe a movement in threedimensional space unambiguously, seven parameters are necessary: three for the position, three for the orientation and a seventh as a path parameter, which establishes the connection between the pure geometric path and its temporal execution [24]. However, ignoring the parameter 'time' and purely planning the path to be followed geometrically, one speaks of path planning. In order to keep the definition of a path as general as possible and initially independent of the robot and tool used, it is generally planned as a movement of the so-called tool center point (TCP), i.e., the coordinate system relevant for the execution of the specific task, in relation to the robot base [25]. According to Craig, the planning of movements in joint space is not critical, since singularities are simply avoided [25]. In contrast, Cartesian planning often has to deal with challenges such as the unreachability of intermediate points, high joint velocities at singularities or the reachability of consecutive poses in different configurations [25]. However, many robotic tasks are bound to fixed Cartesian paths. These tasks include painting, cleaning, inspecting, welding or gluing. The corresponding path planning tasks are closely linked to the challenges of AM. For example, during painting or inspection, a path must be planned that covers the surface of an object without gaps [26]. Consequently, requirements for a path planner can be derived from robotics that complement the process requirements of AM. Specifically, it must be ensured that a planned path can be continuously executed by the robot in velocity limits and without configuration changes. Conventional geometric path planners of AM do not take this into account. Especially for paths that have to be executed in MDAM using WAAM and eccentric wire feeding generated by standard AM path planners, there are high speeds in the joints of the robot to be expected. The planned path has a direct influence on the velocity profile of the joints. Consequently, it should be taken into account already in the path planning step.

3.3. Requirements on New Path Planning Algorithm

The path planning for AM processes and the algorithms used are characterized by the material build-up process. This proves to be sensible, as optimal component quality is the goal of AM. The extrusion process in the FDM process and the optimal setting of the welding parameters in WAAM, in conjunction with suitable path planning, have the greatest influence on component quality. Ding et al. [27] highlighted key challenges for path planning when performing additive welding processes. The robustness of the planning algorithm must be ensured in order to be able to automatically plan paths in complex geometries. WAAM processes pose a particular challenge in this respect, since the width of the material fed and thus the width of the path is large in comparison to the slice area to be filled, unlike in comparable AM processes. Conventional path planners are therefore found to be unsuitable as they do not meet the subsequent requirements. A complete coverage of the slice through path planning must be guaranteed. Non-uniform weld bed profiles can occur over the entire path and lead to accumulated defects in the course of the process. The greatest non-uniformities occur at the beginning and end of a weld path [28]. It can therefore be deduced that a path planning algorithm should produce a continuous path whenever possible. Ding et al. further highlighted path crossing as a challenge [27]. Since crossovers in the manufacturing process lead to undesired material accumulation and stress induction [29], a path planner should avoid them if possible. The analysis of whether a crossover is preferable to setting down and restarting the weld path remains open. These challenges do not lose their importance in the specific case of MDAM. However, the implementation brings the kinematic and dynamic properties of the robot movement and their effects on the welding result to the fore. MDAM opens up a multitude of freedoms and possibilities that enable complex path planning for AM. However, the robot is subject to natural limitations. For example, it must be taken into account that paths planned on a geometric basis in the component coordinate system usually lead to robot movements with high velocities and accelerations, or the robot must adopt configurations that are not achievable from a design point of view or lead to collisions with itself and the environment. Manipulator- and process-specific path planning should consider requirements from both robotics and AM together in one path planning algorithm. The following requirements are to be defined for the path planner:

- Process slices in any spatial orientation.
- Consideration of robot kinematics, joint limits and corresponding assurance of feasibility.
- Geometrical outward approximation of the cross-section.
- Search for the best possible overall path based on defined criteria (from robotics and welding process).
- Minimize calculation time.
- Cover the entire slice without gaps.
- Include path elements that favor continuous and uniform trajectory execution by the robot.
- Modular implementation (extensibility and interchangeability of components).

It can be stated that path planning algorithms essentially perform the optimization within a single slice in order to improve, for example, mechanical properties. Position and orientation of the slice or kinematic movements are not taken into account. The integration of the kinematic system into the path planning as well as a targeted path smoothing are only introduced in the preliminary work of this contribution. The planning of a completely six-dimensional manufacturing process does not exist.

It must be noted that the purely geometric path planning algorithms of the currently pure 2.5D processes are not excluded from an extension to 5D or 6D. The identified deficits in the state of the art are therefore only the lack of consideration of the specific robot used in the manufacturing process as well as the targeted path smoothing and thus are the motivation for the presented development.

4. Concept of Manipulator-Specific Path Planning

The 'divide and conquer' approach was chosen for the concept of a manipulatorspecific path planner. The trade-offs of guaranteeing executability, optimality of robot execution, print result and computational effort can be addressed in this way.

The newly developed path planner first fragments a slice into exclusively convex polygons whose neighborhood relations are recorded in a graph. For each of the convex polygons, multiple infill patterns with varying orientations can be feasible for covering the respective polygon completely. This results in several possible infill options for each convex area, extending the graph. Infills of a polygon are therefore clustered, which means that graph edges can only be generated between infills of distinct clusters. An adapted Hamiltonian search finds the most cost-effective path through all infill options while making sure that every convex polygon is only visited once. In order to avoid calculating a multitude of exact paths that are not relevant to the final path as well as the corresponding robot movement to generate the costs for all possible infill options, simplified evaluation strategies were developed that allow an efficient and fast cost estimation to complete the graph with edge weights. The resulting meta path can then be planned in detail by the infill algorithms used. Furthermore, continuous connections between neighboring infill options are added whenever possible so that the whole slice area is covered by a precisely defined geometric path. The print path is then prepared for trajectory planning by means of smoothing strategies as well as the completion of the path by the print head orientation (Figure 6).



Figure 6. Decomposition path planning algorithm for MDAM.

5. Methods and Algorithms

Before a path can be planned for any component, it must first be sliced, i.e., divided into slices of defined thickness (Figure 7b). The possibility of MDAM and thus the change of building direction can lead to the definition of different subvolumes, where each of them is assigned a distinct building direction based on its shape and relation to the rest of the object (Figure 8). The intersection points of the object's STL representation with a plane result in the polygonal cross-section of the component along this plane. These intersection points (describing the slice polygons) and the corresponding normal vectors of all slices are transferred to the path planner. Each slice is then represented by a collection of line definitions, each connecting two of the intersection points. A slice can contain a single or several closed line loops (polygons), depending on whether the slice consists of a simple surface or contains holes. However, several independently existing islands are not allowed and must be defined in distinct slices.



Figure 7. Approximation and expansion of the slice contour: (a) top view; and (b) side view.

Before dividing a slice into purely convex partial polygons, the outer contours are first simplified. This reduces the calculation complexity in the case of curved outlines and is permissible with regard to the WAAM printing process [30]. Since subsequent post-processing is common with metal parts, the approximated contours are then expanded to ensure that the simplified contour includes the original one completely (Figure 7a).



Figure 8. The result of the multidirectional slicing algorithm.

5.1. Slice Decomposition

After ensuring that the simplified representation contains the original slice completely, it is divided into convex sub-polygons. Convex polygons offer the advantage of allowing the continuous coverage of the polygon area with all infill strategies, especially rasterand zigzag-type infills, independently of their orientation. There are various approaches available for dividing a planar polygon into segments, where each leads to different results [31]. For this work, the approach presented by Ding et al. [21] was chosen to achieve an optimal decomposition of the slice into a minimum number of convex partial polygons. The method can basically be divided into five steps which are executed in a loop until a final set of convex polygons covering the approximated slice has been generated:

- 1. Identify external (outer contour) and internal (holes) polygons.
- 2. Sort the vertices of external polygons clockwise and the vertices of internal polygons counter-clockwise.
- 3. Calculate the angles between the incoming and outgoing edges of each polygon vertex P_i .
- 4. Identify notches. A notch is defined as a vertex whose associated angle from the previous step is greater than 0°. A polygon with at least one notch is not convex.
- 5. Eliminate notches.

While the solution of Steps 1–4 is trivial, the main part of the work takes place in Step 5 of the loop. First, the identified notches are sorted by their calculated angles. The vertices P_2 , P_{10} , P_6 and P_5 represent the identified and ordered notches for the first decomposition loop of the polygon shown in Figure 9. The notch with the largest angle is then selected as the so-called reference notch (here, P_2). Next, the two polygon edges bordering the reference notch are extended until they create the intersection points A_i and B_i (A_2 and B_2) with another edge. Lastly, a distinction must be made among three different cases for creating a dividing line:

- Notch-line (*NL*) decomposition, see $\overline{P_{10}C_{10}}$;
- Notch-vertex (*NV*) decomposition, see $\overline{P_6P_{12}}$; and
- Notch-notch (*NN*) decomposition, see $\overline{P_2P_5}$.

The decomposition algorithm results in connected convex sub-polygons dividing the approximated slice (cf. Figure 9). These convex polygons are then to be filled with standard infill strategies, whereby the infill strategy orientations are free to choose. The strategy itself and its orientation have an influence on the executability of the printing path by the robot. This should be exploited accordingly [6,32].



Figure 9. Slice decomposition algorithm.

5.2. Infill Strategies

Within the scope of this contribution, four different infill patterns are analyzed (see Figure 10): standard-zigzag, raster, enclosed-zigzag and defined-end-zigzag. The raster is composed of parallel lines with identical printing direction where the path is interrupted between each straight segment. The standard-zigzag connects the parallel lines of the raster, creating a zigzag pattern with alternating printing direction. The enclosed-zigzag initially covers the contour of the polygon and subsequently fills the inside of the polygon with the standard-zigzag pattern. The defined-end-zigzag pattern is based on the standard-zigzag pattern including the possibility to select a defined start and end point in the polygon.



(a) Standard-zigzag.

(b) Raster.

(c) Enclosed-zigzag.

(d) Defined-end.

Figure 10. Analyzed infill strategies.

From the point of view of the object to be printed, the impact of geometric deviations to be expected varies and is to be weighted according to the application. Cavities can cause internal weaknesses in the workpiece that can hardly be corrected afterwards. Volume build up, on the other hand, can be corrected, within limits, by subtractive machining. In any case, knowledge of the location of the expected deviation enables potential correction via welding process parameters (see Figure 13).

For each pattern, a main orientation can be identified. It is defined through the start point of the pattern as well as its parallel main lines. The main orientation is then defined by the plane of the infill through being perpendicular to the main infill lines as well as pointing away from the start vertex (cf. Figure 11).



Figure 11. Main orientation (red) of two zigzag infills.

To analyze the quality of each infill pattern within a given area, a coverage analysis is conducted. This analyses the actual coverage of the desired polygon shape through the material output along the calculated geometric path. A geometric approximation is made, expanding each path line segment to a rectangular area of defined width d_{path} and length $l_{segment}$. Points along the path where the welding process starts or ends are additionally marked with an octagonal shape of radius d_{path} (see Figure 12). By mapping these shapes onto a matrix grid with a defined resolution, it can be counted how many times a grid part has been covered by the calculated path.

This approach was already used by Reisgen et al. [33] for measuring the workpiece height with subsequent height control. The actual measured robot path was superimposed on a grid to store the measured part heights. In the case of a 2.5D setup, the workpiece height can be directly correlated with the path coverage. By comparing the coverage analysis presented here with the measured coverage or part height of a real object, weld seam modeling can be improved. Furthermore, a predictive control is conceivable, which does not react to a varying workpiece geometry in the next layer, as described by Reisgen et al. [33]. With this predictive control, the workpiece height could ultimately be ensured in the first layer through dynamic process parameter adaption.

Comparing this matrix M_{path} with the grid matrix M_{slice} of equal resolution and size, which contains the mapped area that the original slice covers, areas with too much material (*coverage* > 1), covered with material (*coverage* = 1) or not covered with material (coverage = 0) can be identified. The analysis provides information about material placed both inside and outside the slice contour. Coverage is captured using three different indices. k_{tot} is defined as the percentage of the intended slice area covered by the path. k_{area} is defined as the ratio of total covered area to total intended slice area. k_{vol} is defined as the added material volume divided by the designed material volume.



Figure 12. Two path segments (left); a path interruption (middle); and coverage calculation grid (right).

All four infill strategies shown in Figure 10 were tested on three test slices: a rectangle, a triangle and a U-shaped slice. The results are shown in Table 2. Exemplary results of the analysis of the rectangle are presented in Figure 13.



(a) Standard-zigzag.

(b) Raster.

(d) Defined-end.

Figure 13. Coverage analysis of exemplary infill strategies of a 80×120 mm rectangle: path width, 10 mm; grid spacing, 0.2 mm; light blue, cavity; pink, material outside of the intended slice; green, one material layer; yellow, two material layers; red, three material layers; purple, four or more material layers.

Table 2. k_{tot} , k_{area} and k_{vol} for all infill strategies and test slices for $d_{path} = 10$ mm.

	Rectangle		Triangle		U				
Infill Strategy	k _{tot}	k _{area}	k _{vol}	k _{tot}	k _{area}	k _{vol}	k _{tot}	k _{area}	k _{vol}
Standard-zigzag	0.96	0.98	1.05	0.86	0.95	1.14	0.80	0.81	0.93
Raster	0.99	1.08	1.37	0.56	0.64	0.86	0.73	0.79	1.06
Enclosed-zigzag	0.96	0.97	1.06	0.93	0.97	1.41	0.91	0.94	1.72
Defined-end-zigzag	0.95	0.97	1.15	0.82	0.90	1.17	0.85	0.88	1.23

5.3. Infill Strategy Weighting

The infill strategies presented can be used in convex polygons (see Figure 14a) in a simple and standardized way. Moreover, in addition to their effect on coverage, they also have an impact on the robot's motion. The graph-based approach analyzes and plans the sub-polygons separately and then, if possible, links them into a continuous path. Each of the polygons can therefore be defined with different infill strategies, different infill orientations and different start and end points. Each individual variation for a single polygon is called an infill object. For each of the identified convex polygons of a decomposed slice, a defined number of infill objects can be created, depending on the infill strategy used. Using a defined-end-zigzag pattern as an example, the number of infill options depends on the number of directly adjacent sub-polygons λ and can be calculated according to Formula(1):

shows the worst coverage abilities of all analyzed infill strategies.

#Infill-Objects(
$$\lambda$$
) = 4 · λ + 8 · $\frac{\lambda \cdot (\lambda - 1)}{2}$ = 4 · λ^2 (1)

The first section $4 \cdot \lambda$ refers to the four possible start- and end-infills. Two adjacent polygons have exactly two common vertices, so each of them can be used as the start (end) of an arbitrary path of which the end (start) point is not important at all. The second section $8 \cdot \frac{\lambda \cdot (\lambda - 1)}{2}$ represents the amount of all possible connection-infills which connect any two common vertices of the given polygon with a different neighbor polygon. The weight that is individually assigned to each infill object is theoretically composed of any number of independent components. At first, the reachability by the robot is evaluated as well as geometric properties of the polygon. In addition to these two aspects, other properties such as the amount of process heat generated along the path and the orientation of the ground cable fixed to the workpiece can be included for each infill object. However, the main objective is to generate a reachable and thus executable path for the robot. All weight components are combined in g_{final} (Formula (2)).

$$g_{final} = g_{reachability} + g_{geometry} (+g_{heat} + g_{cable} + ...)$$
 (2)

Reachability is ensured by checking the reachability of a set of characteristic poses for each sub-polygon and infill object, respectively. First, a reference pose is determined. The position of this pose corresponds to the centroid of the polygon analyzed. The orientation of the corresponding pose is determined by a transformation of the printing plate reference frame according to the following rules:

- The x-axis corresponds to the main infill direction to be analyzed.
- The z-axis corresponds to the layer normal vector.
- The y-axis completes the right-handed coordinate system.

With the help of an already existing package for solving inverse kinematics in the software framework ROS (Robot Operating System [34]), a solution close to a predefined robot configuration is searched for the reference pose. Only if a valid set of joint angles φ_i exists, the full set of characteristic poses is checked. It consists of poses that are composed of all polygon vertices as well as the centroid of the surface. However, three different poses are generated from each polygon vertex by varying the reference frame orientation. In addition to the orientation described for the reference pose (x-axis in infill main direction), two more poses are generated for each vertex, in which the x-axis is rotated around the z-axis by +90° and -90° , respectively, to check both extreme orientations for a standard zigzag type infill (see Figure 15).



Figure 14. (**a**) Coverage graph with 4 nodes connected by 3 edges and (**b**) weighted infill graph of the example slice with 82 nodes and 544 edges.

From the up to eight unique solutions of a standard industrial robot possible for each pose, the one with the smallest Euclidean distance to the reference pose within the joint space is selected. This is done under the assumption that two TCP poses adjacent in Cartesian space belong to the same robot configuration exactly when a direct move from one TCP pose to another causes a minimal movement in joint space. This procedure covers the maximum movement of the robot required for a zigzag infill pattern and ensures that the individual poses are as close to each other as possible and result in as minimal movement of the robot as possible.



Figure 15. Inverse kinematics test-poses for a convex sub-polygon.

The set of different robot poses is next analyzed to calculate the reachability component $g_{reachability}$ (see Formula (2)), if all tested characteristic poses have at least one valid solution. For each robot joint *i*, the maximum motion range $\Delta \varphi_i$ in order to reach all characteristic poses is determined first. A linear function is then used to interpolate proportionally between the minimum ($\Delta \varphi_i = 0$) and maximum ($\Delta \varphi_i = \Delta \varphi_{lim}$) motion (see Figure 16a). To get the complete reachability weight of a polygon, all joint specific weights $g_{reach,i}$ are summed up. For the second part of the infill weight the geometric properties $g_{geometry}$ of the polygon relative to the infill main orientation are analyzed. The maximum length of the polygon along the main direction δ_{length} , as well as the maximum width perpendicular to the infill main direction δ_{width} can be calculated. The ratio $r_{geometry} = \delta_{width}/\delta_{length}$ determines whether many small zigzag movements are necessary to fill the area ($r_{geometry} < 1$), or the path consists of fewer long segments ($r_{geometry} > 1$). A quadratic function depending on a specified critical ratio r_{crit} is used to determine a scalar geometry weight between the

minimum and maximum weight limits (see Figure 16b). In addition, it is also checked at this point whether the analyzed slice allows the planning of a valid path at all. If, for example, the calculated width δ_{width} of the polygon perpendicular to the main infill direction is smaller than twice the expected path width of the welding process, no zigzag-shaped path can be generated at all. By adjusting the limit intervals of the reachability or geometry weight and modifying the interpolation functions, it is possible–after extensive software tests and welding process analysis which are still to be done—to adjust the individual magnitude of the components.



Figure 16. Weights calculation of: (a) *g*_{reachability}; and (b) *g*_{geometry}.

5.4. Meta Path Search

The different weights for each infill option are combined in the infill graph and a least-cost Hamiltonian path is searched for in the graph using a clustered Hamiltonian path search. The starting point of the search is a virtual node that is linked to all infills. This ensures that all infills are represented as start nodes in the edge-weighted graph. Figure 14b shows the infill graph of an example slice. In this three-dimensional way, the different infills per polygon can be visualized. Only those infills that are executable by the robot ($g_{total} > 0$) are shown.

The clustered Hamiltonian path search is tailored to a graph whose nodes can be assigned to smaller clusters. In the context of the path planner, each node maps to individual infill options. These infills can be directly assigned to a geometric sub-polygon of the slice. Connections exist only between infills that do not belong to the same sub-polygon (cluster) and are assigned with weights. The Hamiltonian search is basically NP-hard [35]. In the graphs presented, the computation of a Hamiltonian path would take a significant amount of time. For the special case that each of the polygons can only be represented by a single infill in the final path, the length of the final path and thus the computational effort is reduced considerably. In the algorithm presented, a modified backtracking algorithm and a modified genetic algorithm are used, depending on the number of polygons present within the infill graph. The result of the calculations represents the meta path, which is a sequence of polygons to be traversed with explicitly defined infills (type and orientation).

5.5. Path Planning and Trajectory-Pre-Processing

Having ensured that the necessary joint angles of the path are reachable by the robot, the question appears if a sequence of joint angles can be executed at a given printing speed. This question is referred to manipulability and is treated within this section. The corners in a perfect zigzag pattern require the robot's joint angles to change in an infinitesimally short period of time. Since that is physically impossible, the quasi unsmoothed zigzag path is defined as a zigzag path with a 0.1 mm corner radius and is hereinafter used for benchmarking. Furthermore, the index $k_{\omega,i}$ is defined as the maximum occurring angular velocity of a joint divided by the critical angular velocity limit of said joint. A reasonable printing speed for WAAM processes is 14 mm s⁻¹. Using this speed, the $k_{\omega,i}$ values for the quasi unsmoothed path are indicated in Table 3 when filling a 120 mm by 80 mm rectangle with a 10 mm path width. This path is not manipulable because several $k_{\omega,i} > 1$. To ensure

manipulability, the corners need to be smoothed although the slice coverage should not suffer as a result. Therefore, a smoothing algorithm (cf. Figure 17) is introduced which is based on Bézier curves [36] and includes two parameters: a bump factor Φ and a smoothing factor Ψ .



Figure 17. Smoothing algorithm steps.

0.38

11.4

0.70

(Φ,Ψ)	$k_{\omega,1}$	$k_{\omega,2}$	k _{w,3}	$k_{\omega,4}$	$k_{\omega,5}$	
QU	18.5	31.5	8.15	0.05	11.0	
(0,0.1)	1.85	3.13	0.82	0.01	1.09	

Table 3. $k_{\omega,i}$ values for the quasi unsmoothed path and the smoothing limit cases.

0.56

10.1

1.09

Before creating the actual curve, the Bézier supporting points (green in Figure 17) need to be determined. For each set of three consecutive path points (blue in Figure 17), three supporting points are calculated: do this, an isosceles triangle is created first. The first Bézier supporting point is determined by extending the center line of this triangle by $\Phi \cdot d_{path}$. The second and third supporting points lie on the path lines, $\Psi \cdot d_{path}$ from the second path point, although they can never be further away than half of the path segment length. With these supporting points, a Bézier curve is created. Using this algorithm, the limit cases shown in Figure 18 appear.

0.14

2.58

0.33

0.00

0.03

0.00

0.19

4.03

0.34



Figure 18. Smoothing parameter limit cases. The path starting point is indicated as a green square while the path end point as a red circle.

6. Results

(0,1)

(1, 0.1)

(1,1)

In order to analyze the impact on path planning and to evaluate the target executability, this section first analyzes the functionality of the decomposition path planner in terms of reachability, before analyzing the smoothing algorithm and its impact on manipulability and slice coverage.

*k*_{ω,6} 21.8 2.17

0.38

9.82

0.70

6.1. Analysis of Manipulability Weighted Graph Search

Figure 19 shows an example of the validation of the weight calculation in the path planner. For this purpose, the weights $g_{reachability}$ and $g_{geometry}$ were analyzed for infill orientations in polygons with different edge length ratios and orientations on the print platform. The calculated weights were compared with the exact sum of discrete robot movements. It is evident that the simplified weighting tends to be a correct representation of the expected robot movement. Even though the real minimum of the robot movement is not always correctly represented, the proposed weight calculation functions still leads to the selection of an infill with a significantly reduced robot movement compared to the average robot movement of all twelve infill options calculated.



Figure 19. Exemplary infill weights compared with the real robot motion after trajectory planning (cf. Formula (2)).

If all weights are added up, they can be used as edge weights in the graph. The 82 infill weights for the decomposed slice shown in Figure 14 are presented in Figure 20. Using the tailored clustered Hamiltonian search algorithm, a meta path can be planned. As dead ends in the coverage graph are also connected to all other polygons in the infill graph, there are possible meta path solutions leading to multiple sub-paths in one slice. A fixed weight for those node (infill) connections is used to control the possibility of multiple sub-paths (see Figure 21).



Figure 20. Infill weight combination of the example slices in Figure 7. Pre-processing was done only with standard zigzag. The results in heat input weights are equal to zero for the current implementation.

The influence of the individual weights can be varied. For the example in Figure 20, the influence of the reachability weight in particular was chosen to be higher. The average of

the $g_{reachability}$ is 24.71. As can be seen in Figure 20, infill objects with rather low weights in the range of 15–20 or high weights of approximately 60 can be found. Consequently, many infill objects are easily reachable. However, no significant differences can be discerned among them. For this purpose, the polygon-shape weight is added. The weights lie within a range of 1–20. In this way, infill objects of similar reachability can be distinguished once again. All other weights used were initially given less influence. It can be stated that the weighting generates executable paths. The exact tuning of the weight influences is still pending to be systematically carried out and validated with real experiments.



Figure 21. Executable path planned by the decomposition path planner and the smoothing algorithm.

6.2. Analysis of Path Smoothing

The influence of the smoothing parameters on the slice coverage is analyzed in this section. The indices introduced in Section 5.2 are used. As Figure 22 shows, k_{tot} rises asymptotically with rising Φ and falls with rising Ψ . k_{Vol} rises approximately linearly with rising Φ and falls with rising Ψ , especially for $\Psi \in [0.6, 1]$. In addition, the coverage influence of Φ and Ψ is greater with rising path width. Subsequently, the influence of Φ and Ψ on the joint angular velocity is analyzed. Table 3 shows the $k_{\omega,i}$ values for the limit cases ($k_{tot} \approx 1$) in comparison with the quasi unsmoothed path (QU). It is evident that smoothing offers a promising option to reduce the maximum angular joint velocities of the robot.



Figure 22. Influence of Φ and Ψ on k_{tot} (left) and k_{Vol} (right).

However, the table also shows that only the case ($\Phi = 0$, $\Psi = 1$) is manipulable (all $k_{\omega,i} < 1$). Thus, it can be deduced that the manipulability of the path depends on Φ and Ψ . To further analyze the influence of Φ and Ψ on the manipulability, $k_{\omega,i}$ is plotted for all joints of a Kuka KR 6 as a function of Φ and Ψ (cf. Figure 23).



Figure 23. Influence of Φ and Ψ on $k_{\omega,i}$ for all robot joints. The light blue plane shows $k_{\omega,i} = k_{\omega,i,crit} = 1$ under which the path is manipulable.

Figure 23 shows that $k_{\omega,i}$ rises asymptotically with rising Φ and falls asymptotically with rising Ψ , for all joints. Furthermore, it is evident that the relationship between Φ and Ψ decides if the path is manipulable. When viewing the graphs in Figure 23, the manipulable combinations can be separated from the non-manipulable ones by approximately linear inequalities between Φ and Ψ (cf. Figure 24a for $k_{\omega,1}$). These inequalities define the solution set of smoothing parameters which allow a manipulable path. They can be shown as planes in the k_{tot} 3D plot (cf. Figure 24b), defining what combinations are manipulable and allowing for the choice of the combination which generates the greatest slice coverage.



Figure 24. (a) $k_{\omega,1}$ graph shown from above with a green separation line between manipulable (blue points) and non-manipulable (red crosses) combinations of Φ and Ψ . (b) Separation lines for each joint as a plane for determining the manipulable combinations of Φ and Ψ .

Based on this analysis, the optimal smoothing combination for the Kuka KR 6 and the welding path velocity of 14 mm s⁻¹ is selected as $(\Phi, \Psi)^T \approx (0.6, 1)^T$. Figure 21 shows the example slice after planning including a path smoothing with $\Phi = 0.6$ and $\Psi = 1$. It can further be seen that the path is divided into two sub-paths. While the red path only covers one of the four convex partial polygons, the blue one allows for the continuous covering of the remaining three polygons.

7. Discussion

The inclusion of robot kinematics in the path planning algorithm ensures the reachability of the necessary robot poses for the complex 6D printing in the case of MDAM. The selected influences (see Table 4) of the different weightings already provide executable results. The reachability is supplemented by smoothing the generated path, which thus improves the manipulability. It was shown that the smoothing can reduce the peak velocities of the joints to levels below the respective velocity limits (see Table 3). In order to reduce the negative effects of this smoothing algorithm on the material-coverage of the slice, parameters were identified that combine manipulability with good coverage. Those results need to be further investigated during real welding tests. Overall, it can be stated that the ensured reachability in combination with the optimized manipulability can ensure the desired executability. The results published in this work will be combined with a tailored trajectory planner in future research and in addition tested on a real robotic system. In particular, the trajectory planner can have a great influence on the actual robot movements. Consequently, an evaluation of the robot's trajectory is not meaningful within the scope of this contribution. The test setup available at the IGMR (cf. Figure 25) is to be used for this purpose.

Table 4. Weights mean and standard deviation of the example in Figure 20.

	Mean	SD
Reachability	24.71	18.01
Polygon-Shape	3.93	5.76
Cable	5.29	1.31



Figure 25. Prototype of MDAM process at IGMR using the FDM Process and a Laser Line Scanner for further investigation.

Author Contributions: Conceptualization, M.S., S.M. and M.H.; Formal analysis, M.S.; Funding acquisition, B.C.; Investigation, M.S.; Methodology, J.W. and M.G.; Software, M.S., J.W. and M.G.; Supervision, M.S., B.C. and M.H.; Writing—original draft, M.S., J.W. and M.G.; and Writing—review editing, S.M., B.C. and M.H. All authors have read and agreed to the published version of the manuscript. Funding: This research was funded by Deutsche Forschungsgemeinschaft (DFG) grant number 442454814.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to complex integration of the algorithms into an overall process that is not part of this publication. The pure algorithms are not executable without connection to other algorithms. However, the entire software project can be shared on request.

Acknowledgments: We thank Konstantin Recker, Thomas Nowicki, Florian Menz, Danny Flaiz and Mehdi Zouari for their help along the software implementation. We thank Lukas Oster, Kuhshal Parmar and Rahul Sharma for useful discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Reisgen, U.; Sharma, R.; Oster, L. Plasma multiwire technology with alternating wire feed for tailor-made material properties in wire and arc additive manufacturing. *Metals* **2019**, *9*, 745. [CrossRef]
- 2. Detert, T.; Lorenz, M.; Schmitz, M.; Hüsing, M.; Corves, B. Robotergeführte Objektmanipulation für die generative Fertigung. *IFToMM Tagungsband* **2017**. [CrossRef]
- Ding, D.; Pan, Z.; Cuiuri, D.; Li, H.; Larkin, N.; van Duin, S. Multi-direction slicing of STL models for robotic wire-feed additive manufacturing. In *Solid Freeform Fabrication Symposium*; The University of Texas at Austin: Austin, TX, USA, 2015; pp. 1059–1069. Available online: http://utw10945.utweb.utexas.edu/sites/default/files/2015/2015-88-Ding.pdf (accessed on 20 June 2021).
- Corves, B.; Hüsing, M.; Bezrucav, S.; Detert, T.; Lauwigi, J.; Lorenz, M.; Mandischer, N.; Schmitz, M.; Shahidi, A. Robotik 4.0. In *Handbuch Industrie* 4.0: *Recht, Technik, Gesellschaft*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 569–589. ISBN 978-3-662-58474-3.
- Schmitz, M.; Weidemann, C.; Corves, B.; Hüsing, M. Trajectory Planning Strategy for Multidirectional Wire-Arc Additive Manufacturing. In *ROMANSY* 23—*Robot Design, Dynamics and Control*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 467–475. [CrossRef]
- Schmitz, M.; Corves, B.; Hüsing, M. Multidirektionale Additive Fertigung: Motivation, Problemstellung und Strategien im Kontext des Schweißens mit Drahtzuführung. In Proceedings of the Sixth IFToMM D-A-CH Conference 2020, Lienz, Austria, 27–28 February 2020. [CrossRef]
- Zhang, G.Q.; Mondesir, W.; Martinez, C.; Li, X.; Fuhlbrigge, T.A.; Bheda, H. Robotic additive manufacturing along curved surface—A step towards free-form fabrication. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 721–726.
- 8. Fang, H.; Ong, S.; Nee, A. Robot path planning optimization for welding complex joints. *Int. J. Adv. Manuf. Technol.* **2017**, *90*, 3829–3839. [CrossRef]
- 9. Technikmensch. Simplify3D—Alles über den Premium-Slicer für den 3D-Druck. 2019. Available online: https://technikmensch. de/simplify-3d/ (accessed on 20 June 2021).
- 10. Dunlavey, M.R. Efficient polygon-filling algorithms for raster displays. ACM Trans. Graph. (Tog) 1983, 2, 264–273. [CrossRef]
- 11. Park, S.C.; Choi, B.K. Tool-path planning for direction-parallel area milling. Comput.-Aided Des. 2000, 32, 17–25. [CrossRef]
- 12. Huang, W.H. Optimal line-sweep-based decompositions for coverage algorithms. In Proceedings of the 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Seoul, Korea, 21–26 May 2001; Volume 1, pp. 27–32.
- 13. Farouki, R.; Koenig, T.; Tarabanis, K.; Korein, J.; Batchelder, J. Path planning with offset curves for layered fabrication processes. *J. Manuf. Syst.* **1995**, *14*, 355–368. [CrossRef]
- Wang, H.; Stori, J.A. A metric-based approach to 2D tool-path optimization for high-speed machining. In Proceedings of the 2002 ASME International Mechanical Engineering Congress and Exposition, New Orleans, LA, USA, 17–22 November 2002; Volume 3641, pp. 139–148.
- 15. Zhang, J.; Liou, F. Adaptive slicing for a multi-axis laser aided manufacturing process. J. Mech. Des. 2004, 126, 254–261. [CrossRef]
- 16. Jin, G.; Li, W.D.; Gao, L. An adaptive process planning approach of rapid prototyping and manufacturing. *Robot. Comput.-Integr. Manuf.* **2013**, *29*, 23–38. [CrossRef]
- 17. Bertoldi, M.; Yardimci, M.; Pistor, C.; Guceri, S. Domain decomposition and space filling curves in toolpath planning and generation. In *1998 International Solid Freeform Fabrication Symposium*; The University of Texas in Austin: Austin, TX, USA, 1998.
- 18. Ding, D.; Pan, Z.; Cuiuri, D.; Li, H. Wire-feed additive manufacturing of metal components: technologies, developments and future interests. *Int. J. Adv. Manuf. Technol.* **2015**, *81*, 465–481. [CrossRef]
- 19. Wasser, T.; Jayal, A.D.; Pistor, C. Implementation and evaluation of novel buildstyles in fused deposition modeling (FDM). In 1999 International Solid Freeform Fabrication Symposium; The University of Texas at Austin: Austin, TX, USA, 1999.
- 20. Dwivedi, R.; Kovacevic, R. Process planning for multi-directional laser-based direct metal deposition. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* 2005, 219, 695–707. [CrossRef]

- 21. Ding, D.; Pan, Z.S.; Cuiuri, D.; Li, H. A tool-path generation strategy for wire and arc additive manufacturing. *Int. J. Adv. Manuf. Technol.* **2014**, *73*, 173–183. [CrossRef]
- 22. Lin, S.; Xia, L.; Ma, G.; Zhou, S.; Xie, Y.M. A maze-like path generation scheme for fused deposition modeling. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1509–1519. [CrossRef]
- 23. Webpage of Simplify3d. Available online: https://www.simplify3d.com/. (accessed on 20 June 2021).
- 24. Biagiotti, L.; Melchiorri, C. *Trajectory Planning for Automatic Machines and Robots*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
- 25. Craig, J.J. Introduction to Robotics; Pearson Education International: London, UK, 2005.
- 26. Atkar, P.N.; Greenfield, A.; Conner, D.C.; Choset, H.; Rizzi, A.A. Uniform coverage of automotive surface patches. *Int. J. Robot. Res.* 2005, 24, 883–898. [CrossRef]
- 27. Ding, D.; Pan, Z.; Cuiuri, D.; Li, H. A practical path planning methodology for wire and arc additive manufacturing of thin-walled structures. *Robot. Comput.-Integr. Manuf.* **2015**, *34*, 8–19. [CrossRef]
- 28. Martina, F.; Mehnen, J.; Williams, S.W.; Colegrove, P.; Wang, F. Investigation of the benefits of plasma deposition for the additive layer manufacture of Ti–6Al–4V. J. Mater. Process. Technol. 2012, 212, 1377–1386. [CrossRef]
- 29. Mehnen, J.; Ding, J.; Lockett, H.; Kazanas, P. Design for wire and arc additive layer manufacture. In *Global Product Development*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 721–727.
- 30. Livesu, M.; Ellero, S.; Martínez, J.; Lefebvre, S.; Attene, M. From 3D models to 3D prints: An overview of the processing pipeline. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2017; Volume 36, pp. 537–564.
- 31. Keil, J.M. Polygon Decomposition. Handb. Comput. Geom. 2000, 2, 491–518.
- Corves, B.; Hüsing, M.; Wiartalla, J.; Schmitz, M. 7. IFToMM D-A-CH Konferenz 2021, 18./19. Februar 2021. Online-Konferenz. Manipulatorspezifische Pfadplanung für die Multidirektionale Additive Fertigung. Siebte IFToMM D-A-CH Konferenz 2021: 18./19. Februar 2021, Online-Konferenz 2021. Available online: https://doi.org/10.17185/duepublico/74042 (accessed on 20 June 2021)
- Reisgen, U.; Mann, S.; Oster, L.; Lozano, P.; Sharma, R. Study on Workpiece and Welding Torch Height Control for Polydirectional WAAM by Means of Image Processing. In Proceedings of the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019; pp. 6–11. [CrossRef]
- Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5. Available online: http://www.cim.mcgill.ca/~dudek/417/Papers/quigley-icra2009-ros.pdf (accessed on 20 June 2021).
- Rahman, M.S.; Kaykobad, M.; Firoz, J.S. New sufficient conditions for Hamiltonian paths. In Proceedings of the 2012 IEEE 15th International Conference on Computer and Information Technology (ICCIT), Chittagong, Bangladesh, 22–24 December 2012; pp. 21–25.
- 36. Farin, G.E.; Farin, G. Curves and Surfaces for CAGD: A Practical Guide; Morgan Kaufmann: Burlington, MA, USA, 2002.