Solid Freeform Fabrication 2019: Proceedings of the 30th Annual International
Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference

MULTI-MATERIAL PROCESS PLANNING FOR ADDITIVE MANUFACTURING

Steven Patrick*, Andrzej Nycz*, Mark Noakes*

*Oak Ridge National Laboratory, Manufacturing Demonstration Facility, 2370 Cherahala Blvd,
Knoxville, TN 37932

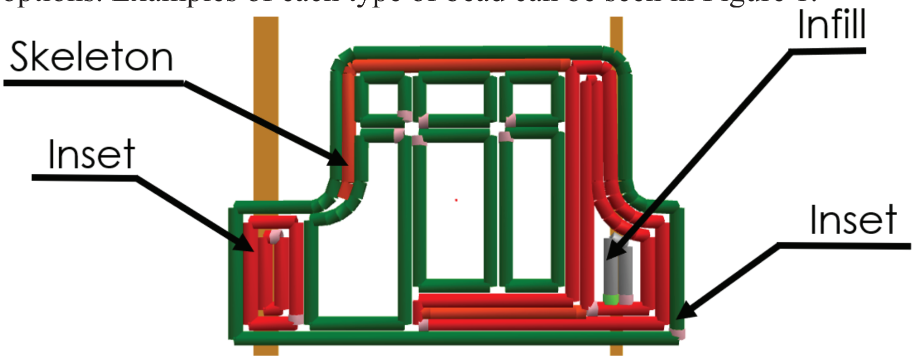
Abstract

A key process in additive manufacturing is converting a 3D model into a set of instructions that a robot can parse and implement. This process is commonly referred to as slicing. Oak Ridge National Laboratory's (ORNL) Metal Big Area Additive Manufacturing (MBAAM) team at the Manufacturing Demonstration Facility (MDF) has developed a slicing software that generates instructions for multiple materials within the same part. The benefits of using multiple materials are lower cost, fewer voids, and greater control over the print. However, a significant challenge arose when the two different materials had different layer heights and bead widths. A layer of complexity was added not only when the material changed from layer to layer, but also when different materials were used within a single layer. These challenges were addressed by assigning bead types and profile types to printing regions and layers, respectively.

Introduction

I. Slicing

In nearly all additive manufacturing processes, the first step is to convert a 3D model into toolpaths a robot can execute to produce a 3D shape out of raw material. This process is referred to as slicing and is executed by a slicer. ORNL has been developing their own slicer for over four years. [1] The majority of slicers create toolpaths by converting a part into stackable layers of a certain thickness. These layers are then put through a 2D path planner in order to fill the desired area. The three main types of generated paths that are discussed in this paper are infill, insets, and skeletons. Insets are defined by closed loop beads that follow the outlines of a part. Infill is a repeatable pattern that is meant to cover a large area. These paths are not closed and do not follow the outline of a geometry. Lines are an example of an infill pattern that consists of parallel lines that are generated a set distance apart until they fill the desired area. Skeletons, which can be defined as an outline following a path that is not a closed loop, are placed wherever insets or infill are not good options. Examples of each type of bead can be seen in Figure 1.



Using multi-material for 3D printing is not a new concept. It has been supported on popular open source slicers, Cura and Slic3r, for many years now. [2] [3] However, these implementations either require the user to split their model into sections or to slice multiple times and combine the results themselves. This is not ideal because both of these processes take extra time for the user to setup the slicing operation. Furthermore, separating the 3D model is not always easily done. There are cases where 3D modelers would not be able to handle it, or it would take an expert modeler to accomplish the task. Clearly, this is not ideal. In addition to these problems, the slicers' path generation is nearly identical for each of the different materials. In MIG welding, the path planning parameters are too variable to treat them as the same material. [4] With all of these problems, it is clear that an automated process for generating toolpaths for multi-material metal 3D printing is necessary.

II. mBAAM System

In order to show the application of multi-material path planning, the MDF's arc system was used. This system consists of a 6DOF ABB arm with a 2DOF positioner. The welders for the system are provided from Lincoln Electric. The dual torch mount was designed in house. MIG welding was used to do additive manufacturing. This type of additive manufacturing is called direct energy deposition (DED). The entire system can be seen in Figure 2.

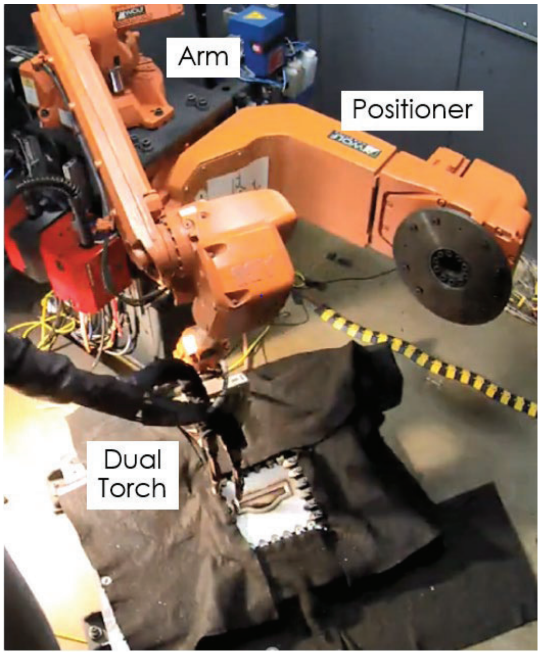


Figure 2. Arc system layout for mBAAM [5]

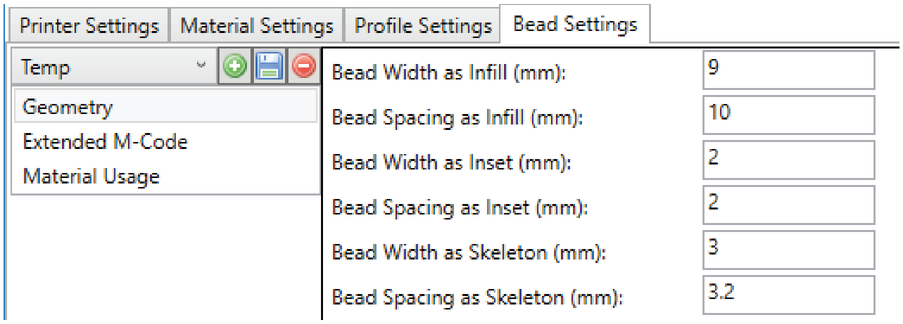
Methods

I. Multi-Material Within a Layer

To print using multiple materials, the user must define the process parameters for successful printing. In ORNL's slicer, these parameters are defined in the material and bead data types. These data types only effect the path planning on a given layer.

A. Bead Type

These data types are defined by the welding material being used. The primary components of beads are the geometry, extended M-codes, and the printing parameters used. Geometry refers to the width and spacing of beads. The slicer has separate values for each type of bead being used in the material. The logic behind this is that different beads serve different functions. Insets are primarily for geometry matching, infill is for rapid filling of a space, and skeletons are meant for hole filling. Therefore, their spacing and widths will differ. The extended M-Codes will be discussed more in Section III.D. The printing parameters are robot commands, like how fast the robot moves and the current used.



Bead Width as Infill (mm): 9

Bead Spacing as Infill (mm): 10

Bead Width as Inset (mm): 2

Bead Spacing as Inset (mm): 2

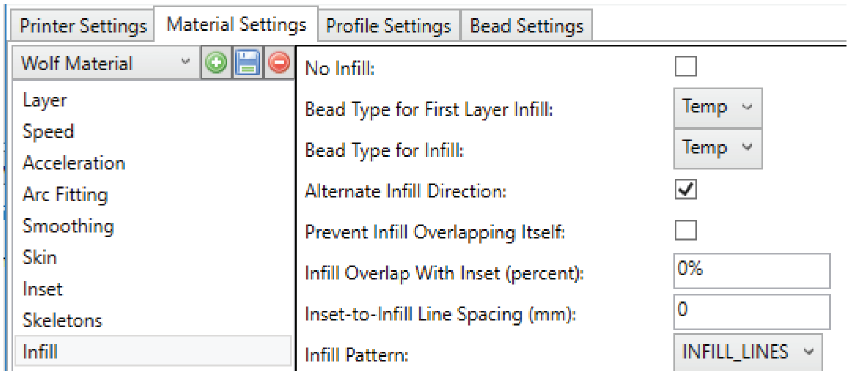
Bead Width as Skeleton (mm): 3

Bead Spacing as Skeleton (mm): 3.2

Figure 2. Example window of bead data entry

B. Material Data Type

A material determines the majority of the path planning attributes. This data type can include the infill pattern, the number of insets, and the minimum path lengths. Another important attribute of the material data is the bead material assigned to the different types of beads. As seen in Figure 4, the bead type for insets was chosen to be Temp. This bead is what was created in the previous section: Figure 3. The infill and skeletons can be another bead type or the same one, depending on what the user desires.



No Infill: □

Bead Type for First Layer Infill: Temp

Bead Type for Infill: Temp

Alternate Infill Direction: ☑

Prevent Infill Overlapping Itself: □

Infill Overlap With Inset (percent): 0%

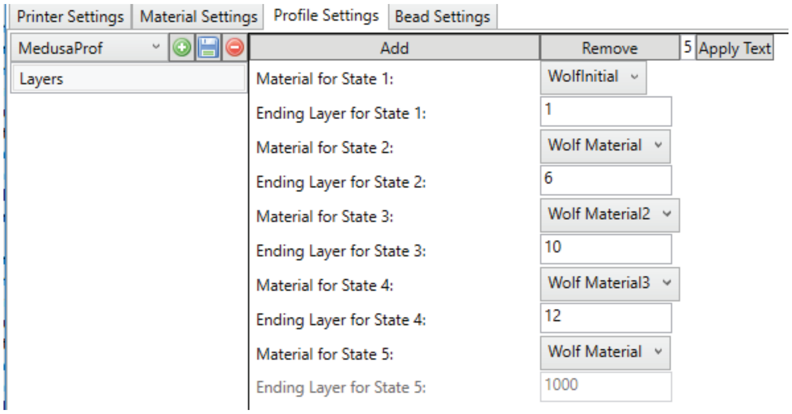
Inset-to-Infill Line Spacing (mm): 0

Infill Pattern: INFILL_LINES

Figure 3. Example window of material data entry

II. Multi-Material Between Layers

In order to have multi-material between layers, a profile data type was created. The only thing contained in this data type is the list of materials used in the print and what layers they apply to. As seen in Figure 5, there are multiple materials used. In addition to multiple materials, there are disjoint layers that have the same material. The implementation of the profile data requires that each material has an ending layer. In the example below, the first layer would be WolfInitial, layers 2-6 would be Wolf Material, layers 7-10 would be Wolf Material2, and so on.



Material for State 1: WolfInitial
Ending Layer for State 1: 1
Material for State 2: Wolf Material
Ending Layer for State 2: 6
Material for State 3: Wolf Material2
Ending Layer for State 3: 10
Material for State 4: Wolf Material3
Ending Layer for State 4: 12
Material for State 5: Wolf Material
Ending Layer for State 5: 1000

Figure 4. Example window of profile data entry

III. Settings Retrieval

In order to accomplish multi-material slicing, all the previously mentioned data types always need to be accessible. Therefore, the layer number must always be known throughout the slice. Before the path planning commences, all the unique material and bead properties are collected and put into a database that the path planner can search. If a material is used for multiple layers, then the material data is only stored once. This helps reduce the amount of time spent searching for printing attributes.

A flow diagram for retrieving a printing attribute can be seen in Figure 6. The first step is to determine if the query is an attribute associated with a layer or the entire print. If it is the latter, then it is considered a printer attribute. These include items like the printer width, height, length or the syntax used for the g-code generation. If the attribute is layer specific, then the profile data type is used. There is only one profile data type for the slicing operation. It retrieves the current layer and finds the corresponding material data type associated with that layer. From there, the query is either a material attribute or a bead attribute. Examples of material attributes could be inset count, infill pattern, and layer height. If it is a material attribute, then the attribute is returned. Otherwise, the appropriate bead data type is found. With that bead data, the appropriate attribute is found and returned. Examples of bead attributes include bead width, bead spacing, and extended M-Code.

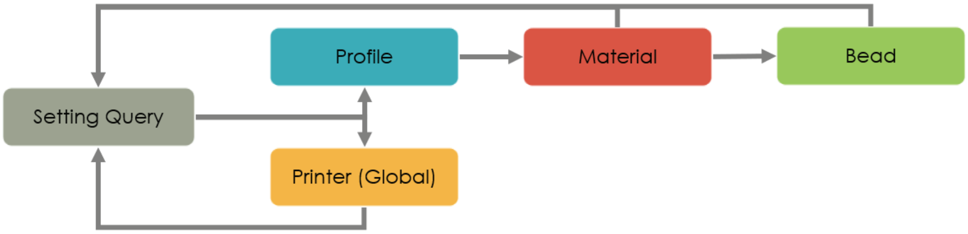
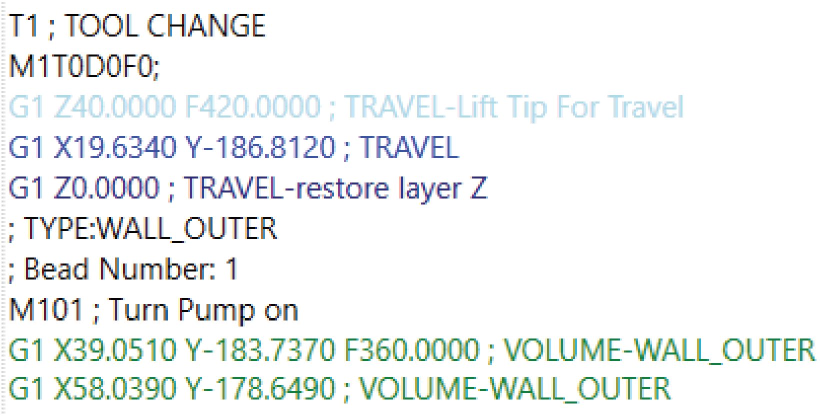


Figure 5. Flow Diagram for accessing printing attributes.

IV. Extended M-Codes

To print using multiple materials, the team had to implement extended M-Codes. These M-Codes were at the start of every bead that was printed. They consist of four attributes: M, T, D, and F. The M attribute describes the type of bead being printed. As mentioned earlier, the options are infill, inset, skeleton. Other options that were not mentioned earlier are pre-starts and tip wipes. These follow the start or end of a bead, respectively, in order to reduce the poor material properties associated with starts and stops. [6] The T attribute describes the tool being used to print the bead. At the MDF, our printer has two tools to print with, so this corresponds to T0 and T1. The next attribute, D, corresponds to the weld mode. Lincoln Electric's welders have many different types of weld modes including spray mode, STT, and power mode. Each weld mode used has a corresponding D code integer. Finally, the F corresponds to the fill percentage for that bead. This controls how thick or thin the bead should be. A value of 5 will produce a half bead and a value of 0 will produce a whole bead. The extended M-code example in Figure 7 can be interpreted as a bead that is an inset, using tool 0, power mode, and printing a full bead.



```
T1 ; TOOL CHANGE
M1T0D0F0;
G1 Z40.0000 F420.0000 ; TRAVEL-Lift Tip For Travel
G1 X19.6340 Y-186.8120 ; TRAVEL
G1 Z0.0000 ; TRAVEL-restore layer Z
; TYPE:WALL_OUTER
; Bead Number: 1
M101 ; Turn Pump on
G1 X39.0510 Y-183.7370 F360.0000 ; VOLUME-WALL_OUTER
G1 X58.0390 Y-178.6490 ; VOLUME-WALL_OUTER
```

Figure 6. Example G-Code with an extend M-Code.

Results

I. Multi-Material Within a Layer

As seen in Figure 8, the infill, insets, and skeletons are all different widths with different spacing. This example is just for demonstration and does not correspond to any printing parameters used at the MDF. Even with this added complexity, the time added to the slicing operation was negligible: less than one second. This is due to the small search space used when searching for a path planning attribute. The number of changes in layer settings or bead data types was always less than ten. This is partially due to not making duplicates in the search space, as mentioned in Section III.C. Therefore, determining the material associated with a layer or the bead associated with an inset, infill, or skeleton takes nanoseconds.

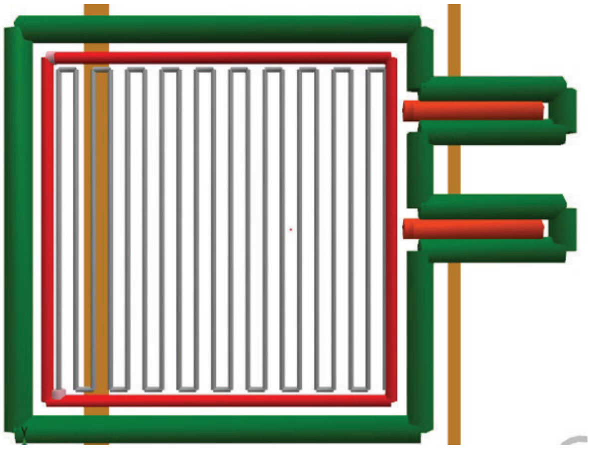


Figure 7. Visual output of ORNL's slicer for multi-material path planning within a layer

II. Multi-Material Between Layers

Just like in the previous section, Figure 9 shows an example of path planning properties changing from layer to layer. The first few layers of the print are much larger than subsequent layers. To highlight the process changing from layer to layer, the top layers have zero insets while the bottom layers have multiple. This shows that the entire path planning operation for each layer is completely independent of previous or subsequent layers. Because of this independence, it is important for the user to look at the output of the slicer to determine whether or not the print will succeed. Things to check for include overhangs and proper filling of each layer.

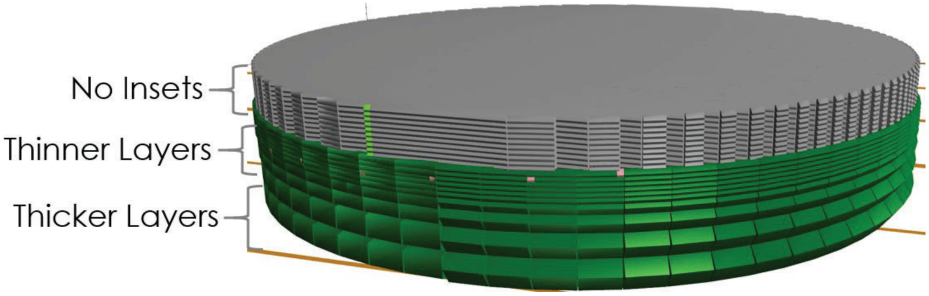


Figure 8. Visual output of ORNL's slicer for multi-material path planning between layers

III. Multi-Material Path Planning of a Real Part

As an example of these methods being used on a real part, the mBAAM team printed a hot stamping die. This print used stainless steel on the outside and mild steel on the inside. More specifically, the skeletons and four insets were stainless steel, and the infill was mild steel. To show between-layer process changes, the initial layer had different material settings than the subsequent layers. This part was over 300 layers and took 9 seconds to slice. As seen in Figure 10, the print was successful.

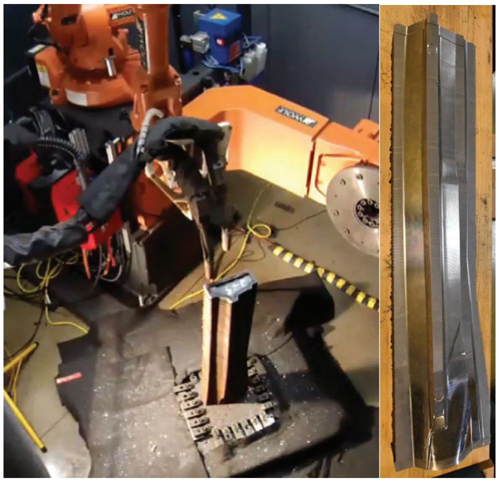


Figure 9. Printing a press using stainless steel and mild steel. Left) Die while being printed. Right) Die after machining

Conclusion

From the results section, it is clear that the multi-material path planning is effective in a real-world scenario. This expands the capabilities of large-scale metal manufacturing with no noticeable addition in slicing time. The algorithms allow parts to be cheaper by only using expensive materials in critical sections. This also can make print times faster because the travel speed of different materials varies. Finally, the user has more control over the material properties of their print. The user can directly specify where certain materials should go in an automated way which was previously not possible for mBAAM processes.

References

[1] A. Roschli, A. Messing, M. Borish, B. Post and L. Love, "ORNL SLICER 2: A NOVEL APPROACH FOR ADDITIVE MANUFACTURING TOOL PATH PLANNING," in *Solid Freeform Fabrication*, Austin, 2017.

[2] Cura, "Cura 13.11.2 User Manual," Ultimaker, 2013.

[3] G. Hodgson, "Slic3r Manual: Multiple Extruders," Slic3r, 2019.

[4] A. Nycz, M. W. Noakes, B. Richardson, A. Messing and B. Post, "CHALLENGES IN MAKING COMPLEX METAL LARGE-SCALE PARTS FOR ADDITIVE MANUFACTURING," in *Solid Freeform Fabrication*, Austin, 2017.

[5] L. J. Love, A. Nycz and M. W. Noakes, "Large Scale Metal Additive Manufacturing with Wolf Robotics," Oak Ridge National Lab, Oak Ridge, TN, 2018.

[6] D. D. e. al, "A tool-path generation strategy for wire and arc," *Int J Adv Manuf Techno,* vol. 73, pp. 173-183, 2014.