

PROGRESS REVIEW 3

Astha Prasad

Team F / ADD_IN

Teammates: Daniel Berman, Nikhil Baheti, Ihsane Debbache

ILR #4

November 12th, 2015

Individual Progress

For this week, my personal goals included the following:

- Display the selected STL file in MATLAB, indicating the insertion layers selected by the user.
- Compile the functional architecture, software subsystem description and status for the PDR.

STL file

In order to plot an STL file, the first step was to understand what it is comprised of. An STL file can be in two formats: Binary and ASCII. They both describe a triangulated 3D part, i.e., they contain the XYZ coordinates of the vertices of every triangle that builds the 3D part. This is accompanied with a unit vector that is normal to the face of the triangle. The direction of the z axis is decided by the right hand rule.

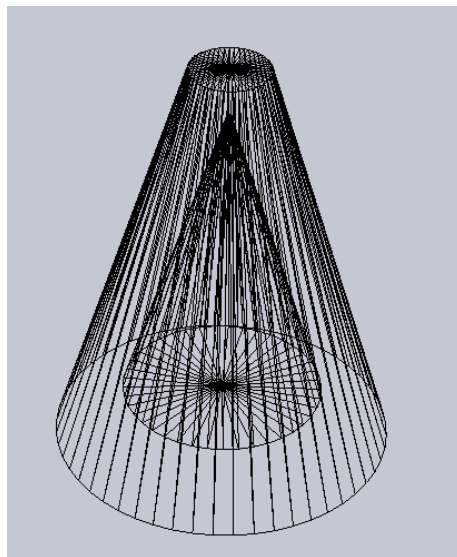


Figure 1 : Screenshot from SolidWorks showing a hollow 3D part broken up into triangles

Format

Every STL file begins with:

```
solid name
```

This is followed by:

```
facet normal -1 4.44089e-11 9.86048e-27
  outer loop
    vertex -3.5 -5e-06 0.5625
    vertex -3.5 5e-06 0.5625
    vertex -3.5 -5e-06 0.4375
  endloop
endfacet
```

The above is repeated for every single triangle present in the 3D part. It lists the normal vector in the first line, followed by the XYZ coordinates of the 3 vertices of the triangle in a loop. After listing these details for all the triangles, the file ends with:

```
endsolid name
```

Plotting

The function to plot the part receives the file name has the following identifier:

```
function PlotSTL(filename,h)
```

It receives the file name as well as the insertion heights provided by the user. In order to plot the STL file, I firstly extracted the XYZ coordinates of each triangle. The MATLAB function 'textscan' makes this process very simple. Since the STL format is uniform for all triangles, the coordinates can be extracted in a loop. They are then plotted in 3D space using the command 'patch(X,Y,Z,D)' where D specifies the colour of the patch. The plotted STL file is depicted in figure 2.

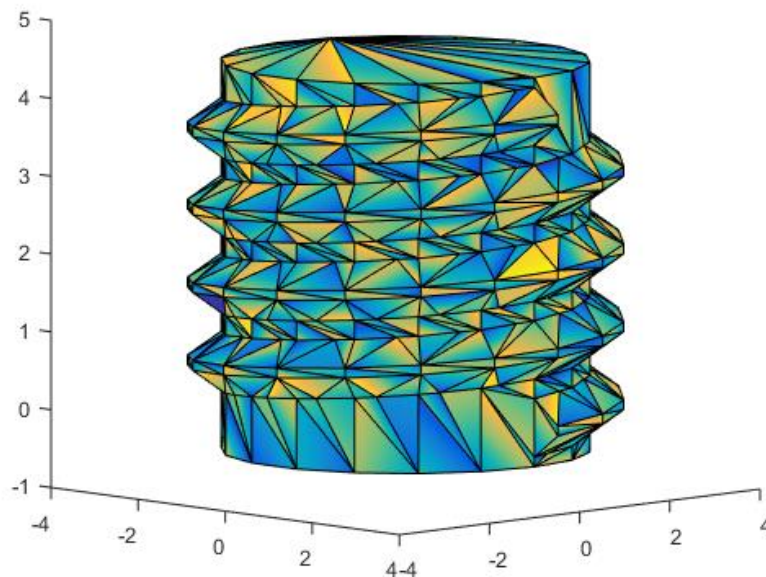


Figure 2 : 3D plot of a screw plotted from its STL file

Once the part has been plotted, the next step is to describe the insertion heights. A patch is plotted at the indicated heights, as is shown in figure 3.

The generated plot will help the user check the insertion heights visually.

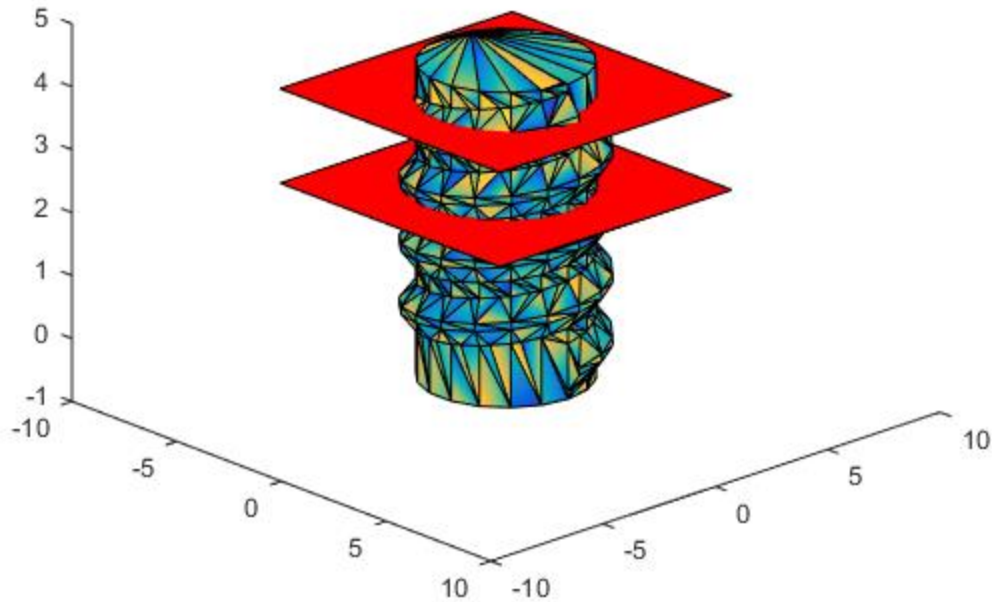


Figure 3 : Patches indicating insertion heights ($h = 2.5$ & 4)

PDR work

For the PDR, I was responsible for the functional architecture and the software subsystem description & current status. Highlights from the above are as follows:

- Since the functional architecture of our project is rather detailed and lengthy, we decided to focus on an overview of the architecture in order to spend less time on this subsection while still communicating how our system functions on a high level.
- The software subsystem description was aimed at highlighted the current approach in implementation, wherein we use MATLAB to invoke Slic3r.

Challenges

A challenge I faced over the last week was understanding what I was expected to present in the functional architecture subsection of the PDR. Since the method of invoking slicer using MATLAB is (hopefully) just temporary, I was unsure if this should reflect in the functional architecture. In order to clear this as well as some more doubts that the rest of the team had, we scheduled a meeting with Professor Dimi to have a discussion about what approach we should follow.

Teamwork

The team split up the tasks in the following manner:

Daniel Berman

Dan took up the task of plotting the g-code of the 3D part. This served as a first step into path planning and 4DOF g-code generation. Currently, the path planning algorithm is such that the

nozzle tip is always oriented in the direction of the COTS item, thus ensuring that the bulk of the nozzle is away from the protruding COTS part. Dan, being the most experienced machinist in the group, machined the new heat block that the hardware team had designed.

Ishane Debbache

Ihsane was responsible for testing the newly machined heat block by installing it into our printer and checking for leaks and inconsistency in its printing capability. I helped him install the new heat block and the entire group was excited to see how the tests with the new bent nozzle went. He was also the one to come up with the idea of using a hollow stepper motor rather than a rotary stage, and did quite a bit of research in this direction.

Nikhil Baheti

Nikhil was in charge of digging into the merlin firmware on our RAMBO board that drives the printer. With a couple of changes to the firmware, he was successfully able to run a 4th stepper motor by parsing g-code through the printer interface software.

Future Goals

Our goals for the next 2 weeks include:

1. Ensure that the filament does not trail as the nozzle goes into insertion configuration.
2. Mount fans onto the printer to ensure better temperature control.
3. Order slip ring and hollow stepper motor
4. Reiterate nozzle design to eliminate leakage of filament
5. Design mount in CAD
6. Drive 4th stepper motor independently by a set distance

I will be focusing on figuring out the correct commands to ensure that the nozzle does not continue extruding filament when it encounters the pause layer.