A SYSTEM AND ARCHITECTURE OF FUSED DEPOSITION MODELING - UNIT MANUFACTURING PROCESS. RAMP CHALLENGE 2017

Arun Bala Subramaniyan* and Rong Pan

School of Computing, Informatics and Decision Systems Engineering

Arizona State University

Email: bsarun@asu.edu*

1 Fused Deposition Modeling UMP

The graphical representation of Fused Deposition Modeling UMP consists of four key elements: Inputs, Outputs, Product and Process Information and Resources. Figure 1.1 shows the graphical representation of FDM UMP model following ASTM E3012-16 guidelines [1]. For a better visualization and detailed understanding, the reader is recommended to refer Input JSON in Appendix A1, Transformation equations in section 2 or Appendix A2, Output JSON in Appendix A3.

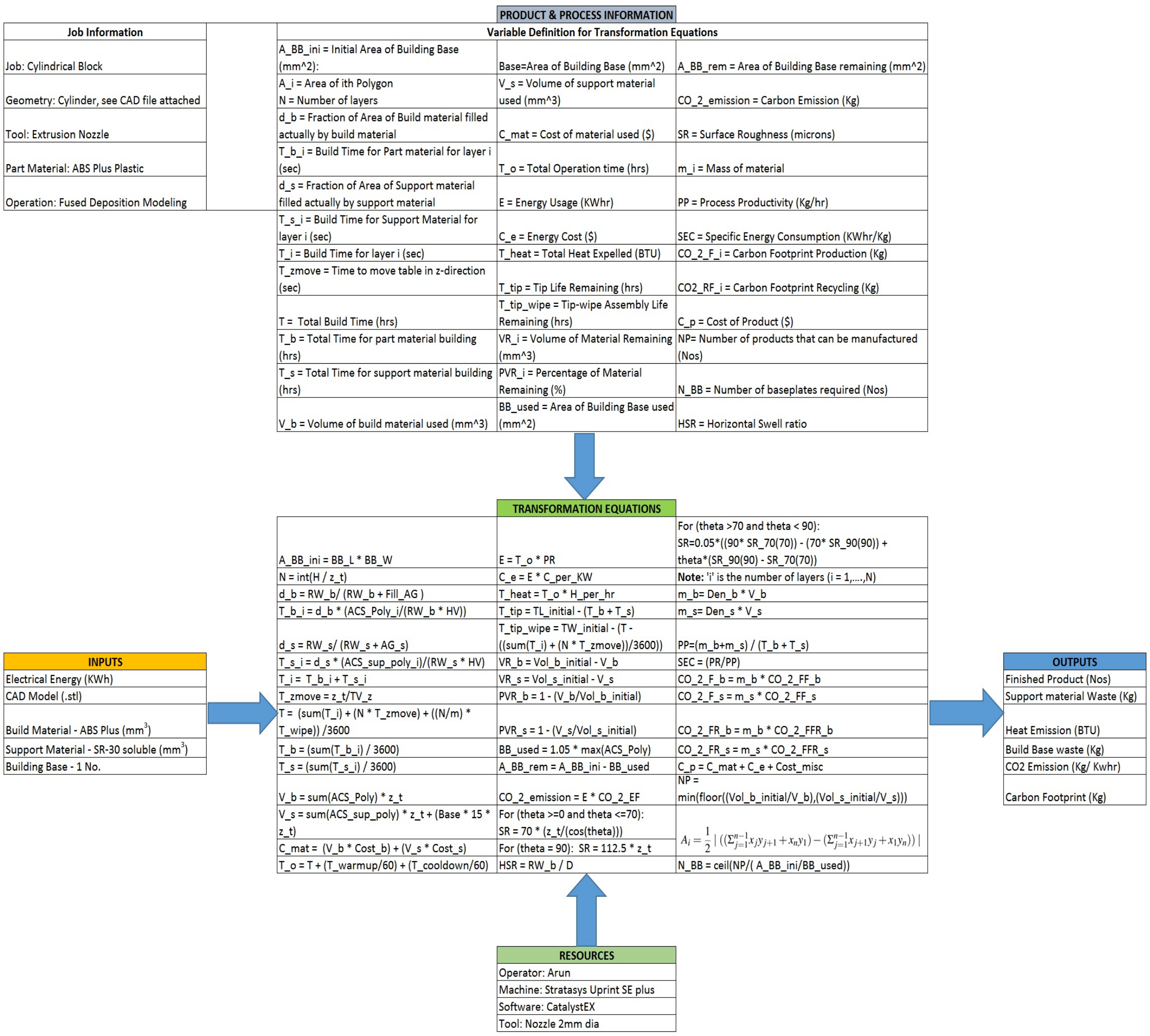


Figure 1.1: Graphical Representation of FDM-UMP

2 Transformation Equations

This section provides the transformation equations that are used for modeling the FDM process. All the equations (2.1 to 2.38) are obtained from references [2]-[14] and description of nomenclature is provided in the next section.

1. Initial Area of Building Base (mm^2) :

$$A_{BB,ini} = BB_l * BB_w \tag{2.1}$$

2. Number of layers :

$$N = \frac{H_{part}}{z_t} \tag{2.2}$$

   where H_{part} is part height calculated from stl data (maximum value of z-coordinate).

3. Area of sectional polygon formed by slicing layer i (mm^2) :

$$A_i = \frac{1}{2} \mid ((\Sigma_{j=1}^{n-1} x_j y_{j+1} + x_n y_1) - (\Sigma_{j=1}^{n-1} x_{j+1} y_j + x_1 y_n)) \mid \quad ;\forall i = 1,\ldots,N \tag{2.3}$$

   where,
   x,y: Vertices of the Polygon (x,y coordinates from stl data).
   n: Number of sides of the polygon (obtained from x,y coordinates).

4. Fraction of Area of Build material filled actually by build material:

$$d_b = \frac{RW_b}{RW_b + Fill_{AG}} \tag{2.4}$$

5. Build Time for Part material for layer i (sec):

$$T_{b_i} = d_b \times \frac{ACS_{Poly_{(i)}}}{RW_b \times HV} \quad ;\forall i = 1,\ldots,N \tag{2.5}$$

6. Fraction of Area of Support material filled actually by support material:

$$d_s = \frac{RW_s}{RW_s + AG_s} \tag{2.6}$$

7. Build Time for Support Material for layer i (sec):

$$T_{s_i} = d_s \times \frac{ACS_{sup-poly_{(i)}}}{RW_s \times HV} \quad ;\forall i = 1,\ldots,N \tag{2.7}$$

8. Build Time for layer i (sec):

$$T_i = T_{b_i} + T_{s_i} \quad ;\forall i = 1,\ldots,N \tag{2.8}$$

9. Time to move table in z-direction for one layer (sec):

$$T_{zmove} = \frac{z_t}{TV_z} \tag{2.9}$$

10. Total Build Time (hrs):

$$T = \frac{\Sigma_{i=1}^{N}(T_i) + (N \times T_{zmove}) + (\frac{N}{m} \times T_{wipe})}{3600} \tag{2.10}$$

11. Total Time for part material building (hrs):

$$T_b = \frac{\Sigma_{i=1}^{N} T_{b_i}}{3600} \tag{2.11}$$

12. Total Time for support material building (hrs):

$$T_s = \frac{\Sigma_{i=1}^{N} T_{s_i}}{3600} \tag{2.12}$$

13. Volume of build material used (mm^3):

$$V_b = d_b(\Sigma_{i=1}^{N} ACS_{Poly(i)} \times z_t) \tag{2.13}$$

14. Volume of support material used (mm^3):

$$V_s = d_s((\Sigma_{i=1}^{N} ACS_{sup-poly(i)} \times z_t) + (10 \times Base \times z_t)) \tag{2.14}$$

    where,
    $Base = ACS_{Poly[max]}$

15. Cost of material used (\$):

$$C_{mat} = \Sigma_i(V_i \times Cost_i) \;\; ;\forall i \in \{b, s\} \tag{2.15}$$

    where,
    b: build material
    s: support material

16. Total Operation time (hrs):

$$T_o = T + \frac{T_{warmup}}{60} + \frac{T_{cooldown}}{60} \tag{2.16}$$

17. Energy Usage (KWhr):

$$E = T_o \times PR \tag{2.17}$$

18. Energy Cost (\$):

$$C_e = E \times C_{perKW} \tag{2.18}$$

19. Total Heat Expelled (BTU):

$$T_{heat} = T_o \times H_{perhr} \tag{2.19}$$

20. Tip Life Remaining (hrs):

$$T_{tip} = TL_{initial} - (T_b + T_s) \tag{2.20}$$

21. Tip-wipe Assembly Life Remaining (hrs):

$$T_{tip-wipe} = TW_{initial} - [T - \frac{(\Sigma_{i=1}^{N}(T_i) + (N \times T_{zmove}))}{3600}] \tag{2.21}$$

22. Volume of Material Remaining (mm^3):

$$VR_i = Vol_{i,initial} - V_i \quad ;\forall i \in \{b, s\} \tag{2.22}$$

where,
b: build material
s: support material

23. Percentage of Material Remaining (%):

$$PVR_i = (1 - (\frac{V_i}{Vol_{i,initial}})) \times 100 \quad ;\forall i \in \{b, s\} \tag{2.23}$$

24. Area of Building Base used (mm^2):

$$BB_{used} = 1.05 * Base \tag{2.24}$$

25. Area of Building Base remaining (mm^2):

$$A_{BB,rem} = A_{BB,ini} - BB_{used} \tag{2.25}$$

26. Carbon Emission (Kg):

$$CO_{2,(emission)} = E \times CO_{2,(EF)} \tag{2.26}$$

27. Surface Roughness (μm):

$$SR = 70 \times \frac{z_t}{Cos(\theta)} \;;\; if \;\; 0^\circ \leq \theta \leq 70^\circ \tag{2.27}$$

$$SR = 112.5 \times z_t \;;\; if \;\; \theta = 90^\circ \tag{2.28}$$

$$SR = \frac{1}{20}(90SR_{70^\circ} - 70SR_{90^\circ} + \theta(SR_{90^\circ} - SR_{70^\circ}))\ ;\ \ if\ \ 70^\circ < \theta < 90^\circ \tag{2.29}$$

28. Mass of Material Consumed (Kg):

$$m_i = Den_i \times V_i \quad ;\forall i \in \{b, s\} \tag{2.30}$$

29. Process Productivity (Kg/hr):

$$PP = \frac{m_b + m_s}{T_b + T_s} \tag{2.31}$$

30. Specific Energy Consumption (KWhr/Kg):

$$SEC = \frac{PR}{PP} \tag{2.32}$$

31. Carbon Footprint for material production (Kg):

$$CO_{2,(F),i} = m_i \times CO_{2,(FF),i} \quad ;\forall i \in \{b, s\} \tag{2.33}$$

32. Carbon Footprint for material recycling (Kg):

$$CO_{2,(FR),i} = m_i \times CO_{2,(FFR),i} \quad ;\forall i \in \{b, s\} \tag{2.34}$$

33. Cost of the product ($):

$$C_p = C_{mat} + C_e + Cost_{misc} \tag{2.35}$$

34. Number of products that can manufactured from the given spool (Nos):

$$NP = \lfloor min(\frac{V_{i,initial}}{V_i}) \rfloor \quad ;\forall i \in \{b, s\} \tag{2.36}$$

35. Number of baseplates required (Nos):

$$N_{BB} = \lceil (\frac{NP}{\frac{A_{BB,ini}}{BB_{used}}}) \rceil \tag{2.37}$$

36. Horizontal Swell Ratio :

$$HSR = \frac{RW_b}{D} \tag{2.38}$$

3 Description of Nomenclature

The nomenclature of variables in FDM-UMP model is given in Tables (1 to 3).

Table 1: Nomenclature of Input keys for Transformation Functions in FDM-UMP

S.No	Name	Description	Unit
1	BB_L	Building Base Length	mm
2	BB_W	Building Base Width	mm
3	BB_H	Maximum Build Height of Machine	mm
4	D	Nozzle Diameter	mm
5	Scale	Model Scale	No.
6	Den_b	Density of part build material	Kg/mm^3
7	Den_s	Density of support material	Kg/mm^3
8	m	Tip wipe factor	No.
9	z_t	Layer Thickness	mm
10	θ	Build Orientation	Degree
11	RW_b	Raster Width of Build Material	mm
12	HV	Extrusion Head Velocity	mm/s
13	RW_s	Raster Width of Support Material	mm
14	$Fill_{AG}$	Model Interior Filling Gap	mm
15	AG_s	Air Gap of Support filling	mm
16	TV_z	Table velocity in Z direction	mm/sec
17	T_{wipe}	Time to wipe the Nozzle	sec
18	$Cost_b$	Cost of Build Material	$\$/mm^3$
19	$Cost_s$	Cost of Support Material	$\$/mm^3$
20	$Cost_{misc}$	Miscellaneous Cost	\$
21	T_{warmup}	Warm up Time	min
22	$T_{cooldown}$	Cool down Time	min
23	PR	Power Rating	KW
24	C_{perKW}	Cost per KW	\$/KW
25	H_{perhr}	Heat Expelled per hour	BTU/hr
26	$TL_{initial}$	Initial Tip Life	hrs
27	$TW_{initial}$	Initial Tip-Wipe Assembly Life	hrs
28	$Vol_{b,initial}$	Initial Volume of Build Material	mm^3
29	$Vol_{s,initial}$	Initial Volume of Support Material	mm^3
30	$A_{BB,ini}$	Initial Area of Building Base	mm^2
31	$CO_{2,(EF)}$	Carbon dioxide Emission Factor	Kg/KWhr
32	$CO_{2,(FF),b}$	Carbon Footprint factor for Build Material Production	Kg/Kg
33	$CO_{2,(FF),s}$	Carbon Footprint factor for Support Material Production	Kg/Kg
34	$CO_{2,(FFR),b}$	Carbon Footprint factor for Build Material Recycling	Kg/Kg
35	$CO_{2,(FFR),s}$	Carbon Footprint factor for Support Material Recycling	Kg/Kg

Table 2: Nomenclature of Output keys for Transformation Functions in FDM-UMP

S.No	Name	Description	Unit
1	V_b	Volume of Build Material Used	mm^3
2	V_s	Volume of Support Material Used	mm^3
3	VR_b	Volume of Build Material Remaining	mm^3
4	VR_s	Volume of Support Material Remaining	mm^3
5	PVR_b	Percentage of Build Material Remaining	%
6	PVR_s	Percentage of Support Material Remaining	%
7	E	Energy Usage	KW
8	$CO_{2,emission}$	Carbon dioxide Emission	Kg
9	SEC	Specific Energy Consumption	KWhr/Kg
10	$CO_{2,(F),b}$	Carbon footprint of Build Material during Production	Kg
11	$CO_{2,(F),s}$	Carbon footprint of Support Material during Production	Kg
12	$CO_{2,(FR),b}$	Carbon footprint of Build Material during Recycling	Kg
13	$CO_{2,(FR),s}$	Carbon footprint of Support Material during Recycling	Kg
14	T_{heat}	Total Heat Dissipated	BTU
15	C_{mat}	Total Cost of Materials (Build + Support)	$
16	C_e	Cost of Energy	$
17	C_p	Total Cost of the Product	$
18	T	Total time to build the part	hrs
19	T_o	Total Operation Time of Machine	hrs
20	T_{tip}	Tip Life Remaining	hrs
21	$T_{tip-wipe}$	Tip-Wipe Assembly Life Remaining	hrs
22	SR	Surface Roughness	μm
23	HSR	Horizontal Swell Ratio	No.
24	NP	Number of products that can be produced from the spool	No.
25	N_{BB}	Number of Building Bases Required	No.
26	m_b	Mass of part build material used	Kg
27	m_s	Mass of support material used	Kg
28	PP	Process Productivity	Kg/hr

Table 3: Nomenclature of Computed Values for FDM-UMP

S.No	Name	Description	Unit
1	$A_{BB,ini}$	Initial Area of Building Base	mm^2
2	A_i	Area of ith Polygon	mm^2
3	H_{part}	Model Height	mm
4	N	Number of layers	No.
5	d_b	Model Interior filling factor for build material	No.
6	T_{b_i}	Time for laying build material for layer i	sec
7	d_s	Support material filling factor	No.
8	T_{s_i}	Time for laying support material for layer i	sec
9	T_i	Build Time for layer i	sec
10	T_{zmove}	Time to move table in z direction	sec
11	T	Total Build Time	hrs
12	T_b	Total Time for part material building	hrs
13	T_s	Total Time for support material building	hrs
14	V_b	Volume of part build material used	mm^3
15	V_s	Volume of support material used	mm^3
16	C_{mat}	Cost of material used	$
17	T_o	Total Operation time	hrs
18	E	Energy Usage	KWhr
19	C_e	Energy Cost	$
20	T_{heat}	Total Heat Expelled	BTU
21	T_{tip}	Tip Life Remaining	hrs
22	$T_{tip-wipe}$	Tip-wipe Assembly Life Remaining	hrs
23	VR_b	Volume of Part Build Material Remaining	mm^3
24	VR_s	Volume of Support Material Remaining	mm^3
25	PVR_b	Percentage of Part Build Material Remaining	%
26	PVR_s	Percentage of Support Material Remaining	%
27	BB_{used}	Area of Building Base used	mm^2
28	$A_{BB,rem}$	Area of Building Base remaining	mm^2
29	$CO_{2,(emission)}$	Carbon dioxide Emission	Kg
30	SR	Surface Roughness	μm
31	m_b	Mass of part build material used	Kg
32	m_s	Mass of support material used	Kg
33	PP	Process Productivity	Kg/hr
34	SEC	Specific Energy Consumption	KWhr/Kg
35	$CO_{2,(F),b}$	Carbon Footprint of build material during production	Kg
36	$CO_{2,(F),s}$	Carbon Footprint of support material during production	Kg
37	$CO_{2,(FR),b}$	Carbon Footprint of build material during recycling	Kg
38	$CO_{2,(FR),s}$	Carbon Footprint of support material during recycling	Kg
39	C_p	Total Cost of the Product	$
40	NP	Number of products that can be produced from the spool	No.
41	N_{BB}	Number of Building based required	No.
42	HSR	Horizontal Swell Ratio	No.

4 FDM-UMP Description

In today's competitive world economy, the manufacturing and design engineers face the challenge of manufacturing components rapidly to meet customer requirements and achieve competitive edge. Additive manufacturing provides an efficient method to build complex products or prototypes to minimize the design and cycle time. Fused Deposition Modeling (FDM) is an additive manufacturing process used to build prototypes using variety of materials. The build and support materials are extruded as a semi-molten filament through the extrusion head and deposited layer by layer to construct prototypes directly from 3D CAD model [2]. This technology is increasingly used for customized products, conceptual models and finds its applications in many fields of engineering and industry like aerospace, automotive products, dentistry and medical implants etc.

When compared to traditional manufacturing processes like milling, drilling, etc., there are not many mathematical abstractions available for characterizing the FDM process. Majority of the literature is focused upon developing models for specific purposes with limited number of parameters, giving insight on how the process behaves with respect to changing parameters and methodology, thereby finding the optimum levels of parameters [2, 3, 4]. In these cases, only the methodology is considered to be useful, as it hinders the use of developed model for different kinds of machines operating under the same FDM principle. So, developing a general abstraction of FDM process is of high importance to the growing user community which helps in saving cost, time and environment, with a major challenge being the limited amount of data and resources. So, in this RAMP-2017 challenge, FDM-UMP model is developed and validated by fabricating a simple component shown in Figure 4.1 with the aid of Stratasys Uprint SE Plus Vertical Machining Center.

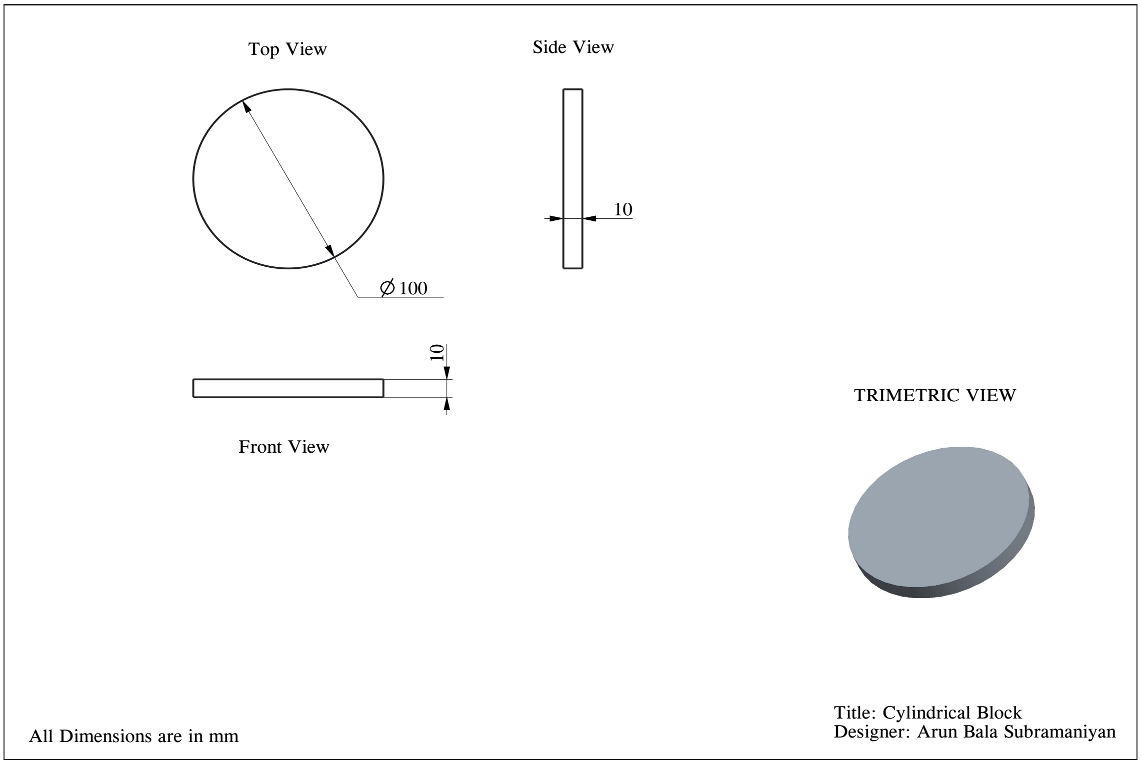
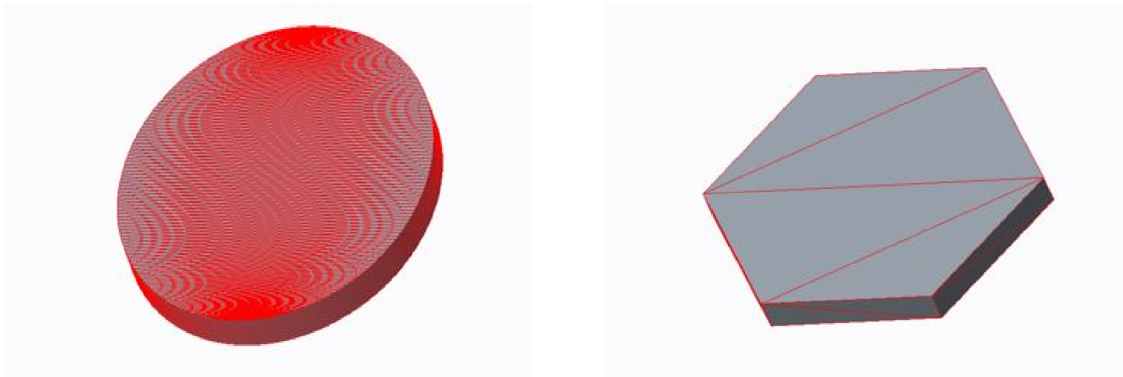


Figure 4.1: CAD Model

The first and foremost requirement for any additive manufacturing process is the CAD model. In FDM, the user uploads CAD model in stereolithography (.stl) format, which is then sliced according to the layer thickness and then prototypes are built layer by layer. So, in this project, the user needs to provide the CAD model in stl format. In addition to stl file, the input parameters are specified as JSON format. The input JSON and stl file of CAD model is read using Python software where all the transformation equations for FDM-UMP are manipulated. Then, the output of python is written to JSON file. The Input JSON, Python code and Output JSON are provided in Appendix and their working is demonstrated in the video (link provided in [15]). The reason that stl file plays a major role in this UMP is that in most situations, the user might not know the actual dimensions of the product, for instance, the CAD designer might be different from the manufacturer. Even, if the user knows the actual part dimensions, the resolution of the stl file affects the final dimension of the product that is manufactured. For instance, the Figures 4.2a (chord length = 0.01mm) and 4.2b (chord length = 10mm) represents two different resolutions of stl file for the same cylindrical block shown in Figure 4.1. Note that Figure 4.2b shows that the circular cross section of original design (in Figure 4.1) will be printed as a hexagon because of change in stl resolution. Hence, the dimensions from stl file will matter the most when compared to actual design dimension for estimating the cost, time and other parameters of FDM. Since the stl file recreates the geometry of the part using a series of polyhedrons (usually triangles) linked to each other, the data from stl file consists of x,y,z coordinates of vertices of all the Polyhedrons that make up the product [7].



(a) High Resolution Stl model: Chord = 0.01mm (b) Low Resolution Stl model: Chord = 10 mm

Figure 4.2: Stl model of cylindrical block in Fig. 4.1 with different chord length

In our model, the z-axis always represents the model height. In the Input JSON, the user specifies all the required parameters like model scale factor, layer thickness etc. The data from stl is sliced according to layer thickness ie., the z-coordinates are sliced into N layers (Equation 2.2). The surface (top-view as shown in Figure 4.1) of each slice will be an irregular polygon depending on the part geometry. The area of irregular polygon is found using Equation 2.3 (Shoelace Theorem [8, 9]). Equation 2.4 provides the filling style for model interior (sparse/dense). If the $Fill_{AG}$ value is set to zero, then it denotes a solid fill whereas, increase in $Fill_{AG}$ will result in sparse filling of model material. The same concept is applicable for support material. Note that the terminologies: part material, build material and part build material are used interchangeably throughout this report. Other Equations 2.4 to 2.13 obtained from [6] are easy to understand. In Equation 2.10, T_{zmove} is the time it takes for the machine to move the table in z-direction after building a layer. Also, in Equation 2.10, the parameter 'm' denotes the frequency at which the tip is wiped (i.e., the tip is wiped once after building m layers). Initially, the FDM machines builds some layer of support material on the base plate in order to set up the base for model building. This information is represented using Equation 2.14. Ten layers of support material based on the maximum area of sliced polygon is assumed to be built before the actual model building takes place but the user can modify the number of initial layers if needed.

Although the layer building is from bottom to top, the layer slicing is done by top-down approach. This will help us to determine the area of support material to be filled. For instance, in top-down approach, when the sliced polygon area of layer 1 is larger than the sliced polygon area of layer 2, then this denotes that there is a need for support material. Because, the machine will build layer 2 first and then layer 1. If there is no support material adjacent to layer 2, then layer 1 cannot be built perfectly. The difference in area of both layer 1 and layer 2 will give the area for the support material to be deposited.

Equations 2.20 and 2.21 can be used to calculate the remaining useful life of the tip as well as tip-wipe assembly respectively. The time taken to build part material and support material is subtracted from the initial tip life, which varies depending on the machine and tip used. Also, the tip is wiped after building some layers. So, the tip-wipe assembly needs to be replaced after a specified amount of time. Equation 2.24 provides the space used by the machine to build part on the building baseplate (BB). An extra 5% of surrounding space is added to accommodate for the distance between parts built on the same base plate.

The surface roughness values in Equations (2.27 to 2.29) depends on the build orientation. The equations provide a good approximation of the surface roughness value as stated in [5, 10] . More accurate value can be obtained if the angle between each face of the polygon and its tangent is known. Some other measures for surface finish and dimensional accuracy like Horizontal Swell Ratio is given in Equation 2.38 [14]. Equation 2.30 provides the mass of material consumed, given by the relationship between mass, density and volume (Density = mass/volume) and the Specific Energy Consumption is provided in Equation 2.32 [11]. The miscellaneous cost (C_{misc}) in Equation 2.35 denotes the cost incurred for stl file, machine operation, baseplates, and hidden costs [12]. The maximum number of products that can be manufactured from the given spool depends on the volume of build or support material usage for each product, (minimum of whichever material runs out first among the two - Equation 2.36). The floor function is used to round the value to integer. Equation 2.37 is the number of baseplates required to manufacture the number of products, obtained from Equation 2.36.

4.1 Validation

In order to validate the developed methodology, the CAD model shown in Figure 4.1 is fabricated (Figures 4.3a and 4.3b). The input parameters are shown in Input JSON code in Appendix A1 and Output JSON is shown in Appendix A3. The input parameters were obtained from machine manual and care was taken to ensure that the machine was not in operation for atleast 24 hours prior to starting the experiments. Since the machine has limitation of choosing only two levels of layer thickness (0.254 mm and 0.33 mm), the experiments are carried out with two levels of layer thickness for two levels of model scaling. The results of the UMP model and actual experimental results are summarized in Tables 4 and 5. The results from the UMP model are acceptable, with some variations occurring due to approximation of unknown input parameters like tip wiping time, table velocity in z direction, frequency of tip wiping, airgap for support and build material, power rating (varies during warm up, machining and cool down) etc. These unknown parameters were approximated using the data from previous experiments. Since, there is no special device to measure most of the output parameters like power consumption, carbon emission, surface roughness etc., only the time (T) and material consumption (V_b, V_s) are measured, using CatalystEX software associated with the Machine.

(a) Isometric View

(b) Top View

Figure 4.3: Fabricated Component

In terms of surface roughness, the value of 20-35 microns is common for parts manufactured by FDM process [5, 10]. The output JSON also shows the value of around 25 microns for all the models based on build orientation of 45°, which is acceptable. In terms of batch processing, the developed method can be used, but with a some modifications. For instance, the time for warmup and cooldown need not be included for each individual product as the machine needs to warmup only once for each batch and the miscellaneous cost (like cost of CAD model, operator cost, machine depreciation, etc [12]) should be taken care in terms of batch, not for a single product.

Table 4: Layer Thickness = 0.254 mm

Part	S.No	Description	FDM-UMP	CatalystEX	Deviation (%)
	1	V_b (mm^3)	78610	80788	2.6
Cyl: (scale = 1)	2	V_s (mm^3)	3108	2949	5.3
	3	T (min)	211	226	6.6
	1	V_b (mm^3)	9856	10159	2.9
Cyl: (scale = 0.5)	2	V_s (mm^3)	760	819	7.2
	3	T (min)	52.6	59	10.8

Table 5: Layer Thickness = 0.33 mm

Part	S.No	Description	FDM-UMP	CatalystEX	Deviation (%)
	1	V_b (mm^3)	78645	80952	2.8
Cyl: (scale = 1)	2	V_s (mm^3)	4028	3605	11.7
	3	T (min)	164	172	4.6
	1	V_b (mm^3)	9863	10159	2.9
Cyl: (scale = 0.5)	2	V_s (mm^3)	975	983	0.8
	3	T (min)	41	46	10.8

Finally, it can be concluded that the developed FDM-UMP model produce acceptable results and the model is easy to use and implement. The requirement for the user is to provide stl file of the part and input parameters of the machine. The running time of the algorithm is fast and will complete in a polynomial time. In our method, all the necessary calculations are done by JSON - Python interface. This model can also be implemented using JSON and JSONiq, but due to advantages of python in reading stl data, this model was developed using JSON-Python interface. Furthermore, this FDM-UMP model can be connected to FDM-cleaning UMP in future to develop a complete system of UMP's for Fused Deposition Modeling process.

References

[1] ASTM International E3012-16 Standard. 2016. "*Standard Guide for Characterizing Environmental Aspects of Manufacturing Processes.*" ASTM International, p:1-10.
`Website: https://www.astm.org/E3012-16.htm`

[2] Grimm, T. 2003. "*Fused Deposition Modeling: A Technology Evaluation.*" Time Compression Technologies, Rapid Prototyping and Manufacturing Association, p:1-12.

[3] Omar, A, M., Syed, H, M., and Jahar, L, B. 2015. "*Optimization of Fused Deposition Modeling Process Parameters: A review of current research and future prospects.*" Advanced Manufacturing, p:42-53.

[4] Omar, A, M., Syed, H, M., and Jahar, L, B. 2015. "*Optimization of Fused Deposition Modeling Process Parameters for dimensional accuracy using I-optimality criterion.*" Measurement, 81, p:174-196.

[5] Thriumurthulu, K., Pandey, P, M., and Reddy, N, V. 2004. "*Optimum part deposition orientation in Fused Deposition Modeling.*" International Journal of Machine Tools and Manufacture, 44, p:585-594.

[6] Alexander, P., Allen, S., and Dutta, D. 1997. "*Part orientation and build cost determination in layered manufacturing.* " Computer Aided Design, 30(5), p:343-356.

[7] Das, P., Chandran, R., Samant, R., and Anand, S. 2015. "*Optimum Part Build Orientation in Additive Manufacturing for Minimizing Part Errors and Support Structures.*" Procedia Manufacturing, 1, p:343-354.

[8] Pure, R., and Durrani,S. 2015. "*Computing Exact Closed-Form Distance Distributions in Arbitrarily Shaped Polygons with Arbitrary Reference Point.*" The Mathematica Journal, 17, p:1-27.

[9] AoPS: Shoelace Theorem,
`Website: http://artofproblemsolving.com/wiki/index.php?title=Shoelace_Theorem`

[10] Ibrahim, D., Ding, S., and Sun,S. 2014. "*Roughness Prediction for FDM Produced Surfaces.*" International Conference Recent treads in Engineering and Technology, p:70 -74.

[11] Baumers, M., Tuck, C, J., Wildman, R., Ashcroft, I., and Hague, R, J, M. 2011. "*Energy inputs to additive manufacturing: Does capacity utilization matter.*" EOS 2011, p:30 - 40.

[12] Mello, C, H, P., Martins, R, C., Parra, B, R. 2010 "*Systematic proposal to calculate price of prototypes manufactured through rapid prototyping an FDM 3D printer in a university lab.*" Rapid Prototyping Journal, 16(6), p:411-416.

[13] Spinnie, N., and Smith, D, E. 2016. "*Large Scale Fused Deposition Modeling: The Effect of Processing Parameters on Bead Geometry.*" Solid Freeform Fabrication 2016: Proceedings of the 26th Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference.

[14] Balogun, V, A., Kirkwood, N., and Mativenga, P, T. 2015. "*Energy consumption and carbon footprint analysis of Fused Deposition Modelling: A case study of RP Stratasys Dimension SST FDM.*" International Journal of Scientific and Engineering Research, p:442-447.

[15] Video Demo of FDM-UMP (Team: Silent Assassins),
`https://youtu.be/IaSZnU1Fvy8`

A Appendix

A.1 Input JSON


```
{
  "ID": "Fused Deposition Modelling",
  "Process_Type": "Unit Manufacturing Process",
  "Input_Parameters": {
    "Build_Material": {
      "Type": "Acrylonitrile Butadiene Styrene (ABS plus)",
      "Filament_diameter": 2.85,
      "Color": "Ivory"
    },
    "Support_Material": {
      "Type": "SR-30 Soluble support (Terpolymer of Methacrylic Acid, Styrene, and Butylacrylate)",
      "Filament_diameter": 2.85,
      "Color": "White"
    },
    "Build_Size": {
      "L": 203,
      "W": 203,
      "H": 152
    },
    "D": 2,
    "z_t": 0.254,
    "theta": 45,
    "RW_b": 1.5,
    "RW_s": 1.5,
    "HV": 25,
    "Fill_AG": 0,
    "AG_s": 5,
    "TV_z": 0.5,
    "T_wipe": 10,
    "Cost_b": 0.0003,
    "Cost_s": 0.0002,
    "T_warmup": 28,
    "T_cooldown": 5,
    "PR": 1,
    "C_per_KW": 0.11,
    "H_per_hr": 2500,
    "TL_initial": 2000,
```

```
        "TW_initial": 500,
        "Vol_b_initial": 491612,
        "Vol_s_initial": 491612,
        "CO_2_EF": 0.4771792,
        "CO_2_FF_b": 0.45,
        "CO_2_FF_s": 0.5,
        "CO_2_FFR_b": 1.3,
        "CO_2_FFR_s": 1.5,
        "Scale": 1,
        "m": 1,
        "Den_b": 0.00000103,
        "Den_s": 0.0000011,
        "Cost_misc": 25
    }
}
```

A.2 Python Code

```
"""
@author: Arun
"""
import numpy as np
import json
import trimesh
import math
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from collections import OrderedDict
with open('Input.json', 'r') as json_data:
     data = json.load(json_data)

BB_L= data['Input_Parameters']['Build_Size']['L']
BB_W= data['Input_Parameters']['Build_Size']['W']
BB_H= data['Input_Parameters']['Build_Size']['H']
D= data['Input_Parameters']['D']
z_t = data['Input_Parameters']['z_t']
theta = data['Input_Parameters']['theta']
RW_b = data['Input_Parameters']['RW_b']
RW_s = data['Input_Parameters']['RW_s']
HV= data['Input_Parameters']['HV']
Fill_AG = data['Input_Parameters']['Fill_AG']
AG_s = data['Input_Parameters']['AG_s']
TV_z = data['Input_Parameters']['TV_z']
T_wipe = data['Input_Parameters']['T_wipe']
Cost_b = data['Input_Parameters']['Cost_b']
Cost_s = data['Input_Parameters']['Cost_s']
T_warmup = data['Input_Parameters']['T_warmup']
T_cooldown = data['Input_Parameters']['T_cooldown']
PR = data['Input_Parameters']['PR']
C_per_KW = data['Input_Parameters']['C_per_KW']
H_per_hr = data['Input_Parameters']['H_per_hr']
TL_initial = data['Input_Parameters']['TL_initial']
TW_initial = data['Input_Parameters']['TW_initial']
Vol_b_initial = data['Input_Parameters']['Vol_b_initial']
Vol_s_initial = data['Input_Parameters']['Vol_s_initial']
CO_2_EF = data['Input_Parameters']['CO_2_EF']
CO_2_FF_b = data['Input_Parameters']['CO_2_FF_b']
CO_2_FF_s = data['Input_Parameters']['CO_2_FF_s']
```

```
CO_2_FFR_b = data['Input_Parameters']['CO_2_FFR_b']
CO_2_FFR_s = data['Input_Parameters']['CO_2_FFR_s']
Scale = data['Input_Parameters']['Scale']
m = data['Input_Parameters']['m']
Den_b = data['Input_Parameters']['Den_b']
Den_s = data['Input_Parameters']['Den_s']
Cost_misc= data['Input_Parameters']['Cost_misc']

#Read Mesh
my_mesh = trimesh.load_mesh('Part.stl')
vert = my_mesh.vertices * (Scale)
#center to origin
gg_0=abs(min(vert[:,0]))
gg_1=abs(min(vert[:,1]))
gg_2=abs(min(vert[:,2]))
vert[:,0]=vert[:,0] + gg_0
vert[:,1]=vert[:,1] + gg_1
vert[:,2]=vert[:,2] + gg_2

#Plot data
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x =vert[:,0]
y =vert[:,1]
z =vert[:,2]
ax.scatter(x, y, z, c='r', marker='o')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
plt.show()

#Layering
H_part = max(vert[:,2])
Layer=z_t
f=[]
P_AB=[]

def cwangle_distance(point):

        vect = [point[0]-origin[0], point[1]-origin[1]]

        lenvect = math.hypot(vect[0], vect[1])
```

```
        if lenvect == 0:
            return -math.pi, 0

        normalize = [vect[0]/lenvect, vect[1]/lenvect]
        dot_prod  = normalize[0]*refvec[0] + normalize[1]*refvec[1]
        diff_prod = refvec[1]*normalize[0] - refvec[0]*normalize[1]
        Ang = math.atan2(diff_prod, dot_prod)

        if Ang < 0:
            return 2*math.pi+Ang, lenvect

        return Ang, lenvect

def PolyArea(x,y):
    return 0.5*np.abs(np.dot(x,np.roll(y,1))-np.dot(y,np.roll(x,1)))

z_t=0

while (H_part/z_t >= 1):
    f=[]
    for i in range(len(vert)):
        if ((vert[i,2]<=(z_t+0.25)) and (vert[i,2]>= (z_t-0.25))):
            f.append(vert[i,(0,1)])
            i=i+1

    f=np.array(f)

    if len(f)!=0:
            pts = f
            origin = [2, 3]
            refvec = [0, 1]
            ff=sorted(pts, key=cwangle_distance)
            ff=np.array(ff)
            x = ff[:,0]
            y = ff[:,1]
            PA = PolyArea(x,y)
            P_AB.append(PA)
```

```
            z_t = z_t + Layer
    else:
        P_AB.append(PA)
        z_t=z_t + Layer


z_t = data['Input_Parameters']['z_t']
P_ABS=[]
a=range(len(P_AB))
for i in reversed(range(len(P_AB))):
    if P_AB[i]>P_AB[i-1]:
        P_ABS.append((P_AB[i]-P_AB[i-1])+P_AB[i])


ACS_Poly = np.array(P_AB)
ACS_sup_poly = np.array(P_ABS)

### Transformation Equations ###

## Initial Area of Building Base $(mm^2)$ :

A_BB_ini = BB_L * BB_W

# Number of layers :

N = int(H / z_t)

## Fraction of Area of Build material filled actually by build
    material:

d_b = RW_b/ (RW_b + Fill_AG )

## Build Time for Part material for layer i $(sec)$:

T_b_i=[]
for i in range(0,N):
    a = d_b * (ACS_Poly[i]/(RW_b * HV))
    T_b_i.append(a)

## Fraction of Area of Support material filled actually by
    support material:

```

```
d_s = RW_s/ (RW_s + AG_s)

## Build Time for Support Material for layer i $(sec)$:

try:
    T_s_i=[]
    for i in range(0,N):
        a = d_s * (ACS_sup_poly[i])/(RW_s * HV)
        T_s_i.append(a)

except IndexError:
    gotdata = 'null'


leng = (N-len(T_s_i))
if leng >0:
        for j in range(0,leng):
            T_s_i.append(0)

#Build Time for layer i $(sec)$:

try:
    T_i=[]
    for j in range(0,N):
        a =  T_b_i[j]+ T_s_i[j]
        T_i.append(a)

except IndexError:
    gotdata = 'null'

# Time to move table in z-direction $(sec)$:

T_zmove = z_t/TV_z

# Total Build Time $(hrs)$:

T =  (sum(T_i) + (N * T_zmove) + ((N/m) * T_wipe)) /3600

#Total Time for part material building  $(hrs)$:

T_b = (sum(T_b_i) / 3600)

```

```
#Total Time for support material building $(hrs)$:

T_s = (sum(T_s_i) / 3600)

# Volume of build material used $(mm^3)$:

V_b = d_b * sum(ACS_Poly) * z_t

#Area of Base $(mm^2)$:

Base=max(ACS_Poly)

# Volume of support material used $(mm^3)$:

V_s = d_s * ((sum(ACS_sup_poly) * z_t) + (10 * Base * z_t))

# Cost of material used (\$):

C_mat =  (V_b * Cost_b) + (V_s * Cost_s)

# Total Operation time (hrs):

T_o = T + (T_warmup/60) + (T_cooldown/60)

# Energy Usage (KWhr):

E = T_o * PR

# Energy Cost (\$):

C_e = E * C_per_KW

# Total Heat Expelled (BTU):

T_heat = T_o * H_per_hr

# Tip Life Remaining (hrs):

T_tip = TL_initial - (T_b + T_s)

# Tip-wipe Assembly Life Remaining (hrs):

```

```
T_tip_wipe = TW_initial - (T - ((sum(T_i) + (N * T_zmove))/3600))

# Volume of Material Remaining $(mm^3)$:

VR_b = Vol_b_initial - V_b
VR_s = Vol_s_initial - V_s

# Percentage of Material Remaining (%):

PVR_b = (1 - (V_b/Vol_b_initial))*100
PVR_s = (1 - (V_s/Vol_s_initial))*100

## Area of Building Base used $(mm^2)$:

BB_used = 1.05 * Base

## Area of Building Base remaining $(mm^2)$:

A_BB_rem = A_BB_ini - BB_used

# Carbon Emission (Kg):

CO_2_emission = E * CO_2_EF

# Surface Roughness ($\mu$m):

def SR_70(theta):
    thet = theta * 0.0174
    SR = 70 * (z_t/(math.cos(thet)))
    return SR

def SR_90(theta):
    SR = 112.6 * z_t
    return SR

def SR_70_90(theta):
    SR=0.05*((90* SR_70(70)) - (70* SR_90(90)) + theta*(SR_90(90)
        - SR_70(70)))
    return SR

if (theta >=0 and theta <=70):
    SR = SR_70(theta)
```

```
elif (theta >70 and theta < 90):
    SR = SR_70_90(theta)
else:
    SR = 112.5 * z_t


# Mass = Density * Volume (KG):

m_b= Den_b * V_b
m_s= Den_s * V_s

#Process Productivity (Kg/hr)

PP=(m_b+m_s) / (T_b + T_s)

# Specific Energy Consumption (KWhr/Kg):

SEC = (PR/PP)

# Carbon Footprint Production (Kg):

CO_2_F_b = m_b * CO_2_FF_b
CO_2_F_s = m_s * CO_2_FF_s

# Carbon Footprint Recycling (Kg):

CO_2_FR_b = m_b * CO_2_FFR_b
CO_2_FR_s = m_s * CO_2_FFR_s

#Cost of Product ($):

C_p = C_mat + C_e + Cost_misc

# Number of products that can be manufactured (Nos):

NP= min(math.floor((Vol_b_initial/V_b)),math.floor((Vol_s_initial
    /V_s)))

# Number of baseplates required (Nos):

N_BB = math.ceil(NP/( A_BB_ini/BB_used))

```

```
# Horizontal Swell ratio:

HSR = RW_b / D

# Write to Output JSON

Output = {
  "ID": "Fused Deposition Modelling - UMP",
  "Output_Parameters": {
    "Material": {
            "V_b": V_b,
            "V_s" : V_s,
            "VR_b": VR_b,
            "VR_s": VR_s,
            "PVR_b": PVR_b,
            "PVR_s": PVR_s,
            "m_b": m_b,
            "m_s": m_s
    },
    "Sustainability": {
          "E": E,
          "CO_2_emission": CO_2_emission,
          "SEC": SEC,
          "CO_2_F_b": CO_2_F_b,
          "CO_2_F_s": CO_2_F_s,
          "CO_2_FR_b": CO_2_FR_b,
          "CO_2_FR_s": CO_2_FR_s,
          "T_heat": T_heat
    },
    "Cost": {
        "C_mat" : C_mat,
        "C_e": C_e,
        "C_p": C_p
    },
    "Time": {
        "T ": T,
        "T_o" : T_o,
        "T_tip": T_tip,
        "T_tip_wipe": T_tip_wipe
    },
    "Miscellaneous": {
          "SR": SR,
```

```
            "HSR": HSR,
            "PP": PP,
            "NP ": NP,
            "N_BB": N_BB

        }
    }
}
with open('Output.json', 'w') as json_output_data:
    json.dump(Output, json_output_data, sort_keys=True)
```

A.3 Output JSON


```
{
  "ID": "Fused Deposition Modelling - UMP",
  "Output_Parameters": {
    "Cost": {
      "C_e": 0.44685559103814804,
      "C_mat": 24.204810458101715,
      "C_p": 49.65166604913986
    },
    "Material": {
      "PVR_b": 84.00973106641678,
      "PVR_s": 99.36760475215051,
      "VR_b": 413001.9190902328,
      "VR_s": 488503.06907414214,
      "V_b": 78610.08090976716,
      "V_s": 3108.9309258578373,
      "m_b": 0.08096838333706018,
      "m_s": 0.0034198240184436213
    },
    "Miscellaneous": {
      "HSR": 0.75,
      "NP ": 6,
      "N_BB": 1,
      "PP": 0.035679096672789457,
      "SR": 25.084632282856337
    },
    "Sustainability": {
      "CO_2_FR_b": 0.10525889833817824,
      "CO_2_FR_s": 0.005129736027665432,
      "CO_2_F_b": 0.03643577250167708,
      "CO_2_F_s": 0.0017099120092218107,
      "CO_2_emission": 1.9384563040646423,
      "E": 4.062323554892255,
      "SEC": 28.0276154178154,
      "T_heat": 10155.808887230638
    },
    "Time": {
      "T ": 3.512323554892255,
      "T_o": 4.062323554892255,
      "T_tip": 1997.634799778441,
      "T_tip_wipe": 498.90833333333336
    }
```

```
    }
}
```