Case Study Additive Manufacturing Project with M290 Printer, Assessing Material Properties of A6061-RAM2, and Exploring Failure-Detection in 3D printing

CU Denver Senior Design

Dr. Jensen, Dr. Selman

Rock Adcock, rock.adcock@ucdenver.edu

Jesus Casillas, (720-481-7785, jesus.casillas@ucdenver.edu)

Samuel Edison, (720-209-7817, samuel.m.edison@gmail.com)

Raphael Hatami, (720-249-9205, raphaelhatamijackson@gmail.com)

Joseph Lavelle, (joseph.lavelle@ucdenver.edu)

Jonathan Malm, (720-483-7460, jonmmalm@gmail.com)

Victor Meraz, (victor.meraz@ucdenver.edu)

Gabriel Piper, (425-478-0254, gabepiper@gmail.com)

Table of Contents:

1. Abstract

3D-printing metal is a concentration of additive manufacturing that is on the cusp of a revolution, and its potential can now be further investigated with emerging technologies recently made available. This includes exploring the capabilities of selective laser melting (SLM), as well as analyzing the still-nascent data on 3D-printed material properties, such as that of A6061-RAM2--a non-standard alloy of aluminum. In this year-long project, we utilized topology optimization and finite element analysis to design, analyze, prototype (via and CAD and PLA printing), and produce a missile interstage component using an M290 metal printer. We then developed a post-processing plan, involving machining, heat treatment, verification tests, etc., to ensure quality and make our part flight-ready. Additionally, we conducted materials testing and analysis including cold and hot tensile tests at extreme temperatures, fatigue testing, surface roughness tests, and frequency tests. Lastly, we developed foundational code and a proof-of-concept for error detection in metal printing applicable to printers such as the M290. We monitored cost-models turn-around times and other factors to assess the economic and expedient quality of our production process from the ideation stage to production. Our results demonstrated that 3D-printing with A6061-RAM2 is a potentially-viable option, particularly in low-quantity parts such as those designed for aerospace, with satisfactory material properties for a wide range of applications. We also developed error detection that could vastly save time, money, and improve the quality of printed parts in any context. From our results, we estimate that given further testing, machine vision application to the printer would likely be able to identify flaws in parts that test coupons would not catch. Furthermore, given more confidence in the results, these techniques could be used to avoid the need to destructively test as many coupons per print to still have confidence in the structural integrity of printed pieces.

2. Overview

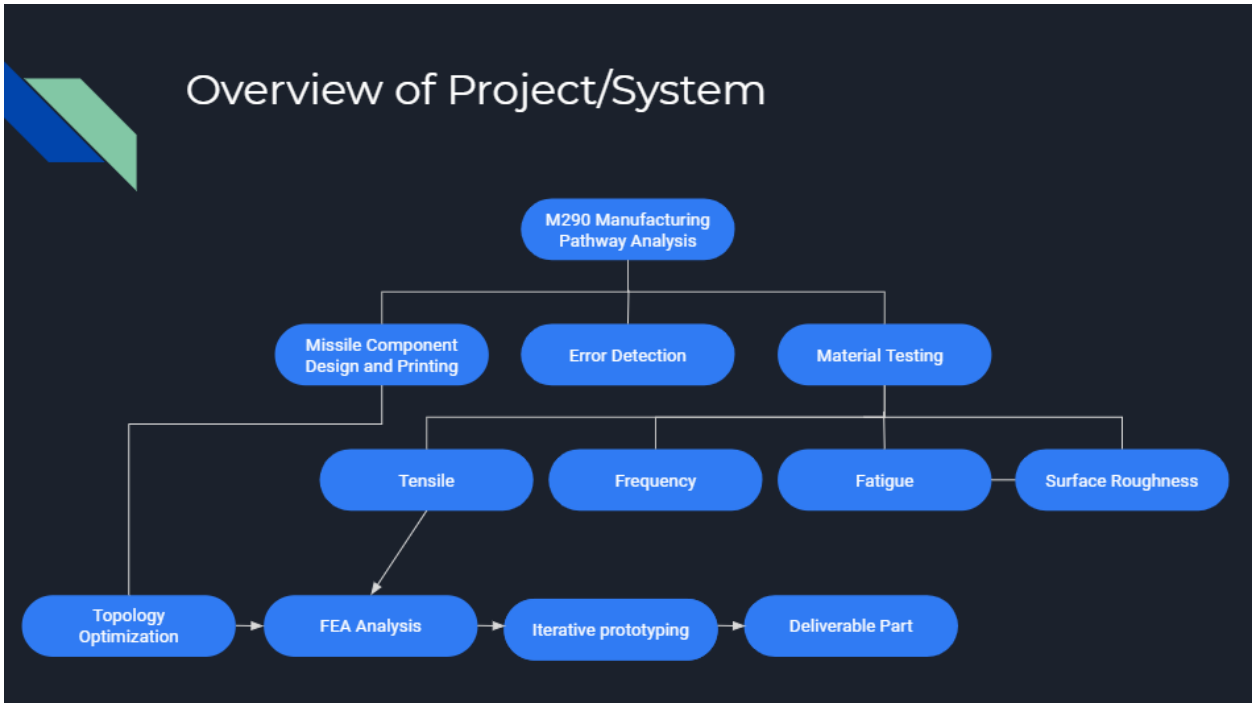


Figure 1. *Simplified flow diagram of our project.*

There are three main problems that this project seeks to address. First, conventional manufacturing of complex parts is costly and time-consuming. The piece we are looking to design is particularly intensive to machine into the desired product. Second, there is a dearth of material data for additively manufactured metals. This is important because additive manufacturing processes differ substantially enough from traditional methods that it cannot be assumed that material properties will be identical. Finally, there is currently not a reliable method of identifying part defects that occur during the recoating phase of the printing process.

In regards to our case study design of a missile interstage component, our generative design process principally considers the aspects of conventional machining that are both time and cost-intensive. Conventional manufacturing (i.e. machining) of complex geometries is labor-intensive and may require breaking the part into sub-parts. As a result, the cost of conventional manufacturing can be quite high. Complex parts that can be printed in one piece

and that need minimal machining will save manufacturers time and money, while allowing them to take advantage of the benefits of complex geometries. New printers such as the M290 are further bringing large-scale and mass production of 3D printed metal components to reality. Such applications are especially pertinent to the aerospace industry.

In regards to our material testing, our complete testing plan involves fatigue testing with as-printed coupons, the results of which will be compared with that of a prior study conducted by Dr. Yakacki to determine how surface finish affects fatigue properties. Additionally, we will complete an extreme temperature tensile (0° C to 230° C) to explore how temperature affects the 3D-printed material's behavior. We will also execute a surface roughness test to further quantify the as-printed coupons as they relate to our fatigue tests, as well as resonance frequency tests.

Regarding our focus in error detection, the additive process can introduce additional errors or weak points that are not present in traditional manufacturing. The M290 printer used in our tests is a powder-bed based additive manufacturing system. This entails the machine laying a thin layer of fine aluminum powder then applying a high powered laser to the desired print area. The main sources of error are either inaccuracies caused by the laser pass or uneven/lack of aluminum powder coating in critical positions of the print. As of the writing of this paper, the laser pass has not been a major source of error. The M290 has built-in infrared cameras that analyze the thermal imaging of the print. The thermal images do such an excellent job of noting any potential failures in the laser pass that the errors detected are rarely looked at due to the incredibly minor faults it detects. On the other hand, the conventional cameras native to the M290 do not have any error detection software at this time. This means that powder distribution is not directly monitored unless human interference is deemed necessary. Early detection of print faults would allow for techs to abort catastrophically flawed prints earlier, saving time and costs. Additionally, a reliable fault detection method would increase confidence in part performance and thus increase the viability of the additive manufacturing process. Increased

confidence in prints could also reduce the number of destructive tests needed to verify the integrity of individual pieces.

3. Stakeholder Needs

The needs of our stakeholders (those at Lockheed Martin) align with the previously stated problems. These needs are primarily: cheaper production, more material property data, and in-process error detection.

The primary customer need is a lower-cost alternative to conventional manufacturing for the production of complex aerospace (or similar) parts. While the missile interstage, along with other aerospace components, are generally produced in low quantity, additive manufacturing can still greatly reduce cost of prototyping as well as cost per unit of final product, which will greatly increase savings. As stated in the problem overview, traditional manufacturing methods for complex geometries are labor-intensive and therefore expensive. Even with the expense of cutting edge metal additive manufacturing processes, the potential reduction of labor costs generated by being able to print complex geometries may offset that expense.

An additional customer need is for more material data for printed aluminum. Of particular interest is performance data of printed materials without a surface finish versus with a surface finish. This question seeks to ask: is surface finishing necessary? If not, this could be a valuable cost and time-saving technique. This relates back to the previous customer need of cheaper production. If unfinished parts can perform as well as finished parts, or at least perform to minimum requirements, there will be even more savings in post-processing labor. If this is found not to be the case, it will still be important for major manufacturers such as Lockheed Martin to know if 3D printed materials behave the same as traditionally manufactured materials in this regard, allowing for better predictions for cost savings later down the production timeline.

Finally, the customer needs to be able to optimize the usage and performance of any 3D printer. Since many products produced by Lockheed Martin are designed for space and other

stressful applications, it is critical to make sure the quality of produced parts is very high, and the functionality well-prevised. Optimizations of print time and material that promote quality are imperative for improving operations.

Our affinity diagram has three categories: Optimized Production, Cheaper Production, and Optimized Performance.

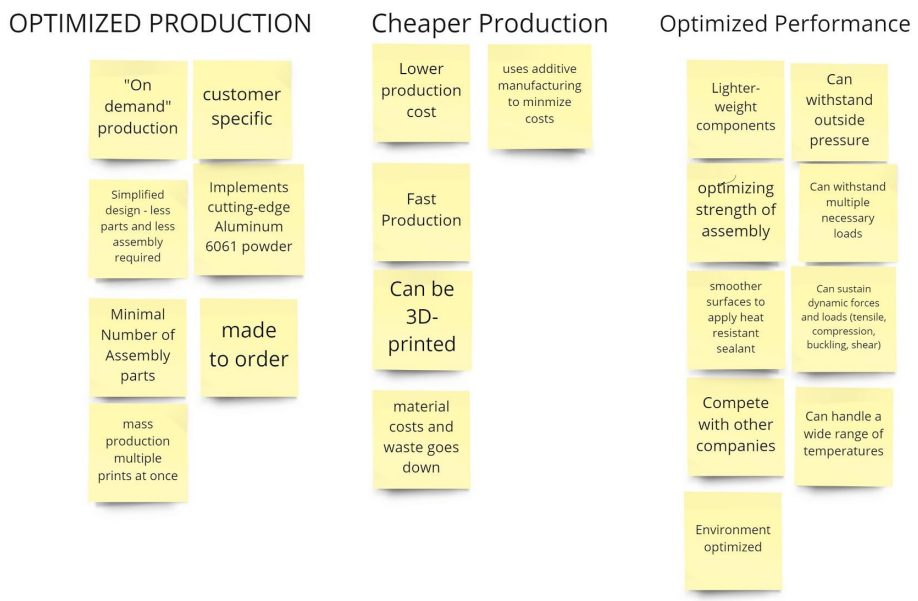


Figure 2: *Affinity diagram*

4. Other Design Methods

Although there are specific ideas related to design, there are also alternative methods that were considered applicable for our design (shown in our journey map in Figure 3). For example, we would assume that there are projects that involve detonating a test artifact, and these projects would need approval from any professional company. One considered idea related to this is to design a missile assembly.

Firstly, the project leads would be informed by their stakeholders/sponsors, i.e. Lockheed Martin, for specific needs, such as specified materials, constraints, etc. The material should then be assessed for affordability and feasibility with respect to manufacturing. Research would also

need to be completed to find resources for producing the product. Before this stage, however, material tests and simulations would need to be performed to construct the best prototype of the product based on the properties and physical conditions required or desired. Of course, if results are unsatisfactory (determined ultimately by stakeholders/sponsors), the process must then be repeated until convergence to a desired qualitative state.

In this project, as soon as we complete testing and studies of the design, the manufacturing timeline will begin. In this case, additive manufacturing will be exploited to the fullest advantage given the low-quantity and design constraints of our part. Using computer vision methods to detect errors, we can make sure we have a higher quality control when printing said parts. After that, we would transport the products to Lockheed Martin to test the part at a missile facility. If they approve the design, then we would assemble the interstate section so we could deliver a completed assembled missile to a customer in need.

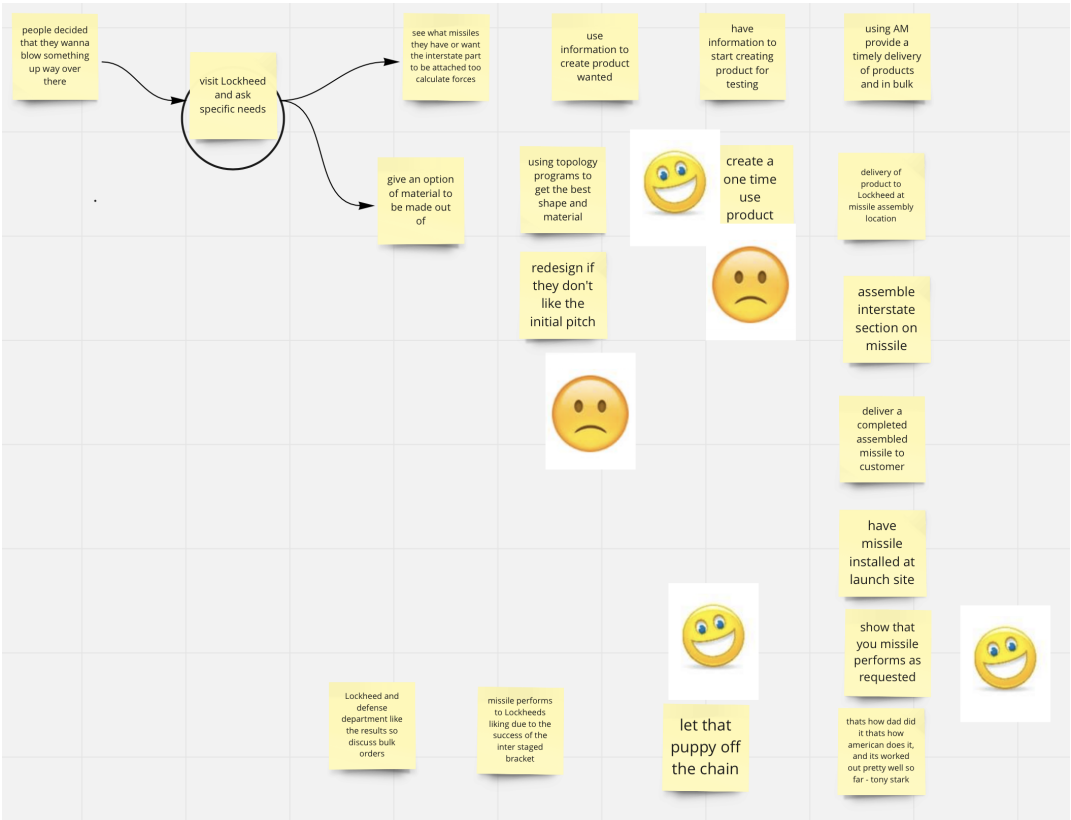


Figure 2: *Journey map + others*

5. Ideation

Note: Principal aspects of this section are gone over in further detail in sections 6 (Design Requirements) and 7 (Analysis/Prototype Plans).

Though many parameters of our design were already provided by Lockheed Martin (see section 6, Design Requirements), our conceptualization and ideation process was still required.

Our journey started with a choice in the project. We then sat down as a team to think of simple ways to map out certain accomplishments to achieve the intended goals. We were given the goal of creating a part using new, partly-unproven technology. The creation of a one-time or several-use product would need to meet the satisfaction of Lockheed Martin. Our journey map laid out a plan to conceptualize, create, and deliver a part and a method to produce and further the utility of manufacturing parts using additive manufacturing. Our journey map was used as an initial tool to connect ideas and processes in a compartmentalized way of conceptualizing. Our Journey map then evolved into a ‘mind map’ of further ideas.

The mind map produced was done so with the thought of creating something using the highest-quality technologies and methods. When considering the production of an interstage missile component, we used the ideas of a lighter stronger part, use of topology optimization software, part simplicity, and material testing to assimilate our technique and approach.

With material testing the idea is to find out as much about the material as possible. The testing of the compression, tension, and tensile strengths of the material will also be beneficial to further study and use. With compression testing done in labs, it was witnessed that previous materials such as polymer prints would perform differently due to internal print patterns used. The possibility of looking into different orientations of parts printed could affect the testing performance. Noting how material is affected by heat and cold treatment could also be worth investigating. We would want to further investigate the transpired-property effect. To make the part lighter there would have to be optimization of the created part. To shortly summarize here

(more detail later), we first started out with a part that was much more in mass and used iterative and generative design (both manually and via CAD) to reduce the part down in weight. Using the mind map we went down and completed the process of using the optimization program to create a thinner and lighter part. The part began off solid unrefined and with different iterations the part improved by maintaining strength and reducing the weight. The internal lattices could be viewed to see changes in parts structure as well. With a view of the possibilities of the product creation, the thought of future plans excites everyone on the team.

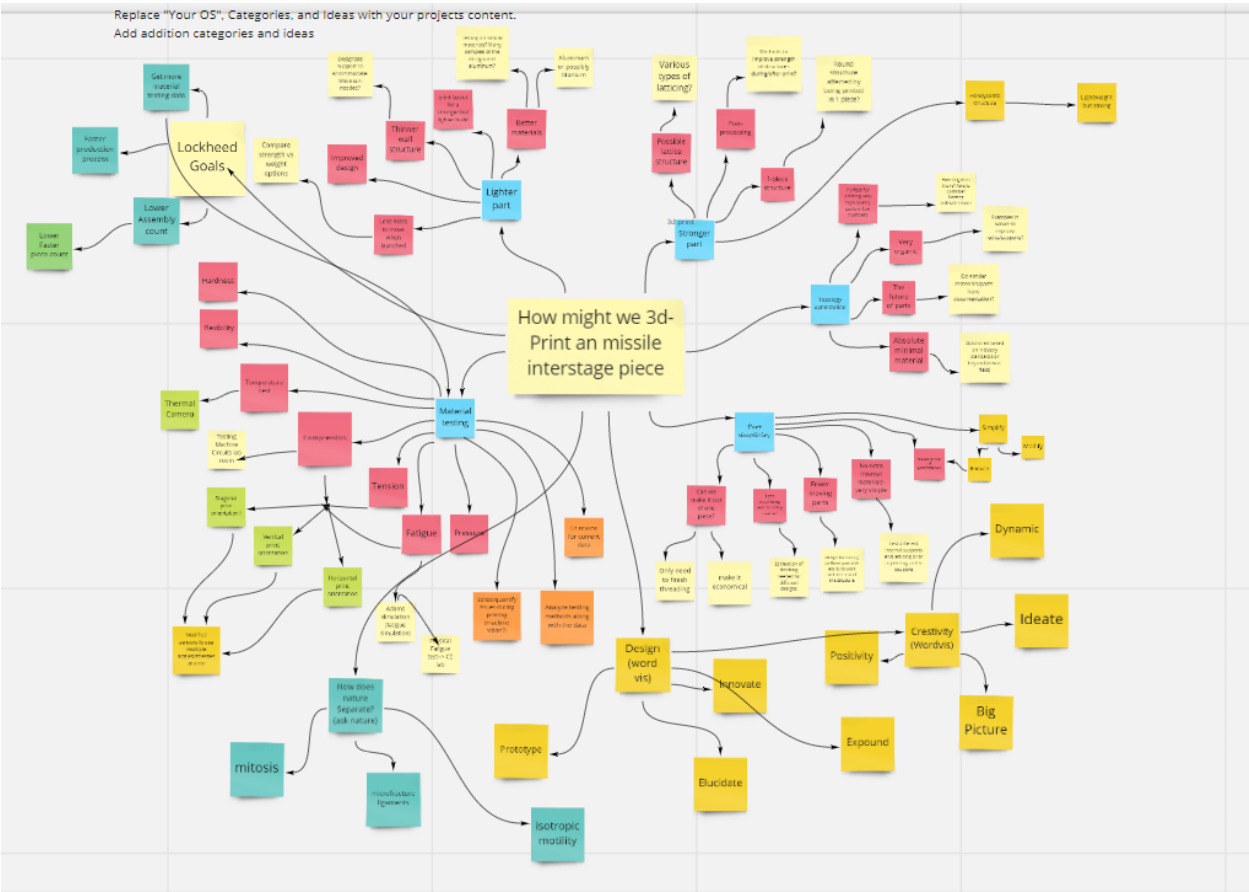


Figure 3: *6-3-5 sketches*

Design By analogy

Another exercise, designed by analogy, was implemented, using AskNature and WordVis. Wordvis provided us with concepts and relations to these concepts that we hadn't yet

perceived. In the AskNature exercise, were examined scenarios in nature that might help us with the internal mountain of Mass 3 (further explained in section 6 of this paper), with analogies of insectoid carapaces and arachnoid exoskeletons researched for inspiration for our design.

Feasibility and Novelty of Concepts Generated

Although given the intrinsic nature of our project in which feasibility must be weighted higher than novelty, we still understood the importance of novelty in generating unforeseen solutions to problems. In total, we generated seven concepts, several of which are given in more detail in the next sections. The novelty of these concepts were initially anemic, although we then explored the importance of novelty while grappling with the ongoing trouble of satisfying our mass 3 requirements (see section 6), and novel ideas relating to machining were also considered. Our final concept and design was eventually narrowed down using the real-win-worth technique--a collaborative exercise in which each team member produced their initial designs given the design requirements (section 6). Our ultimate choice was further narrowed down using this technique in tandem with engineering principals, topology analysis, and faculty consultation.

6. Design Requirements

The design requirements primarily apply to the intersage portion of the project with specifications given to us by the Lockheed Martin team. The initial model was a simple cone-shaped structure that was going to be 12.8” tall and less than 5lbs. The base is 9.8” in diameter, with a top section of 5.286” diameter. The overall height was changed to 11” (Revision B), as the M290 printing machine could not fit the initial height. The external wall of the cone had to remain intact and could not be compromised.

The part will also support three main masses; Mass 1, 5 lbs; Mass 2, 20 lbs; and Mass 3, 2.5 lbs. Mass 1 was located 8” above the top section of the structure. Masses 2 and 3 are to be located inside the part itself. Mass 2 will be bolted up to an internal 6-bolt section, which is 7.25” from the base. Mass 3 will be bolted onto the inner wall of one side of the cone and will be located 2” from the outside plane of the cone, centered at 4.25” from the base.

The part also has specific parameters for bolting everything together. Mass 1 will be connected to the top by using a 6-bolt pattern, #8-32 bolt. The base of the structure will be attached with a 4-bolt pattern using #10-32 bolt. Mass 2 will be attached internally to a 6-bolt section using #8-32 bolts.

Mass 3 will be mounted to the internal wall, located 2” away from that outside wall. The mounting plane will be parallel with the external angle of the cone. The mass will be bolted on a flat, 4”x4” mounting point, and is centered 4.25” above the base of the cone.

The final design is also subject to strength and material property requirements which will allow the interstage to perform as designed under its expected operating conditions. When complete, the interstage must be able to withstand 25 G’s of axial acceleration and 15 G’s of lateral acceleration. The material chosen must be able to withstand the forces associated with those accelerations in an operating temperature range of 32 ℉ to 250 ℉ (0 ℃ - 121 ℃). The final interstage must have a natural frequency of 20Hz or greater. Material testing is being conducted to ensure that the AL6061-RAM2 aluminum powder used to print the interstage has the needed strength characteristics under the expected operating temperatures, and will give the interstage a high enough natural frequency.

Other more general design guidelines, aimed at production and assembly efficiency, have also been considered during the design process. To improve production efficiency and reduce errors during printing, all threaded holes will be designed to be drilled out and tapped during post-processing. To improve assembly efficiency, the interstage will be designed in as

few parts as possible, and require the fewest possible fasteners. For this reason, the interstage will be designed to use single-piece machine screws, rather than bolts that require washers and nuts, or other fastener types that require multiple parts. Ideally, the interstage will be printed as a single piece, and each internal mass will be mounted with the minimum number of machine screws as is safely possible.

7. Analysis/Prototype Plans

A.) Advanced Part Design

The part itself was readily modelled in Solidworks. The biggest challenge will be keeping an internal diameter of 6.7”, so that Mass 2 can be properly installed. This creates multiple issues. The internal attachment of Mass 3 may be difficult with this specification. As the design process progresses, modeling a viable solution to meet both Mass 2 and 3 requirements will cause a continued evolution of the part.

The first steps were to model the part in its basic shape, using Solidworks. Additionally, since the part height requirement was initially too tall to print in the M290 machine, the overall height was adjusted to the ‘Revision B’ of 11”. Our first model (***Figure 4***), was large and bulky. It’s weight was 22 lbs, using stock 6061-T6 aluminum.

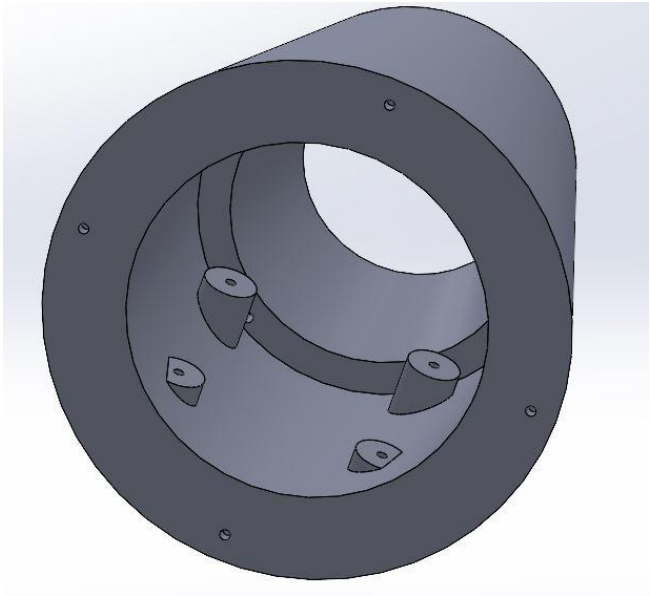


Figure 4: *Initial model (SolidWorks).*

From here, we knew that we had to lighten and redesign the model so that we can meet the desired specifications of less than 5 lbs. Our next step was to use computer simulation software in order to reduce the mass of the structure.

We began using the topology optimization software of nTopology in order to reduce the mass of the part. The simulations removed material from the internal wall, continually reducing mass. Since we had to constrain the outer shell of the cone, the software kept reducing the wall thickness in order to achieve the desired mass.

The decision was then made to use the outer 'shell' of the part, and possibly fill the internal structure with some type of lattice (***Figure 5***) or gyroid (***Figure 6***) design. At first, the idea sounded attractive; we anticipated the ability to reduce the mass of the part, all while adding an intricate internal structure that will be lighter, but just as strong. After consulting our M290 expert, Lockheed Martin's Joe Block, on this idea, he advised us that this would be difficult and challenging.

First, the ability to test the print of these structures is strenuous and difficult. Since the printed metal will be inside of the shell, it would be much more challenging to assess the success of the printed lattice. Secondly, if anything were to break off internally, there would be no practical or worthwhile way of fixing it. Mr. Block strongly suggested a different route, such as possibly using just the external shape of the cone, but reducing material from the inside-out.

From here, we decided to keep things simple. Since we wanted a part that was feasible to test, print, and view, we chose to maintain the outer part of the shell and reduce the internal mass of the bolt flange area, along with the outer wall thickness.

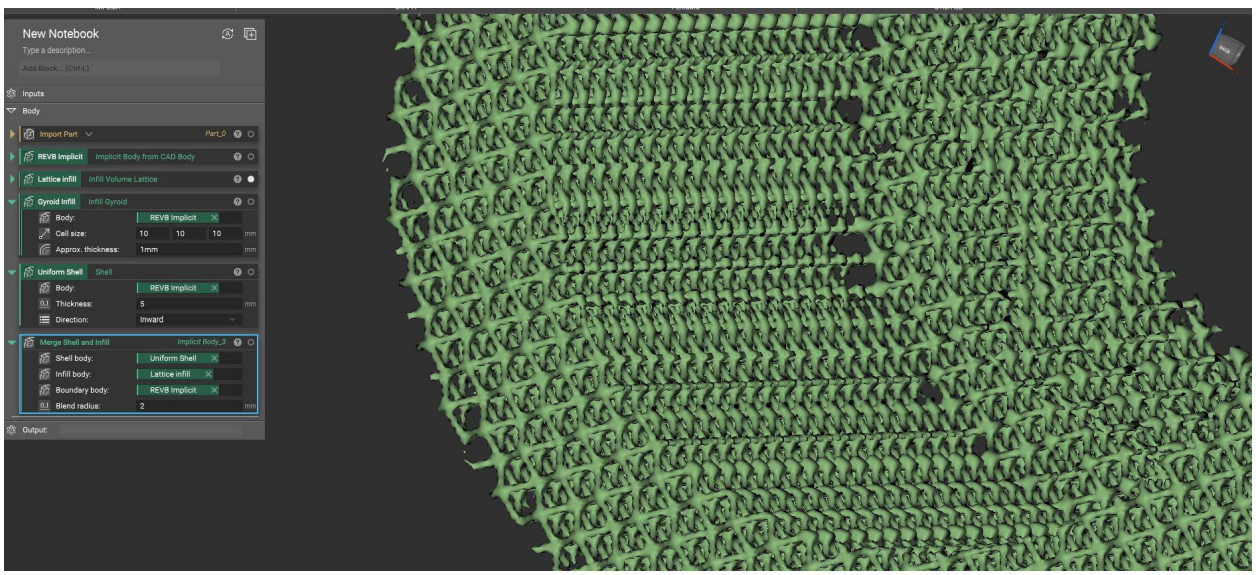


Figure 5: Internal lattice structure

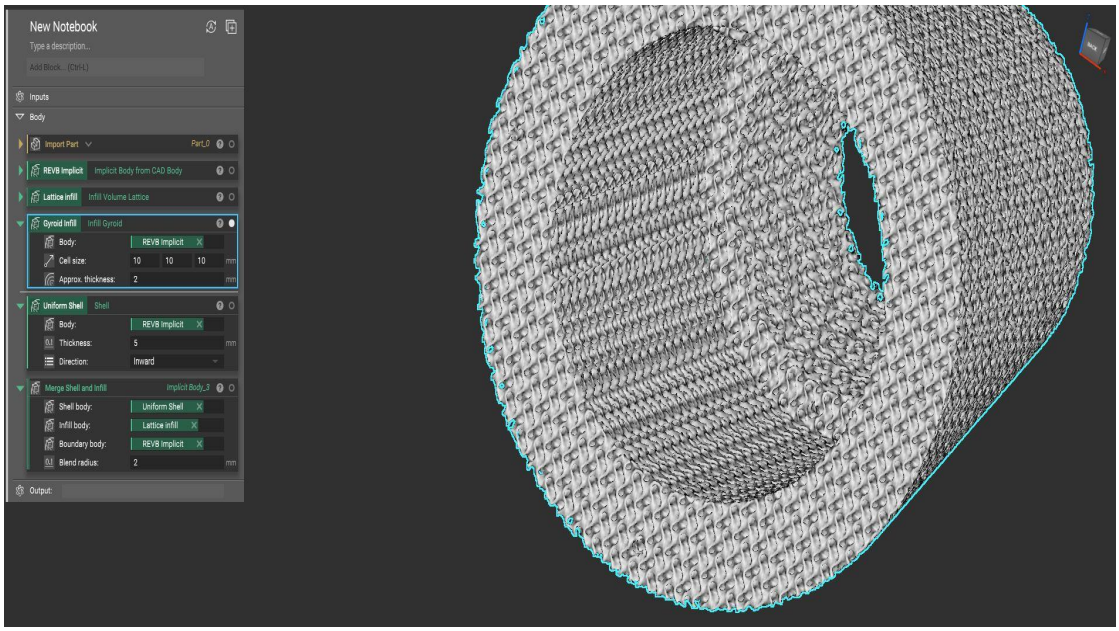


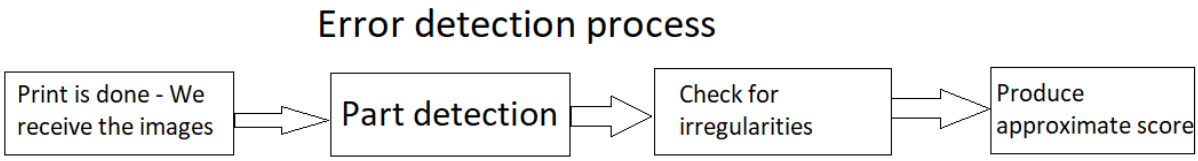
Figure 6: Internal gyroid Structure

B.) Material Testing

Tensile

To determine if the 3D printed metal parts will be strong enough to withstand the forces associated with the axial and lateral accelerations that the interstage will be subject to, within the range of expected operating temperatures, tensile testing of 3D printed metal ‘dog bone’ coupons was conducted to quantify the elastic modulus, yield strength, ultimate strength, and failure strength of the 3D printed metal--A6061-RAM2. The interstage is expected to operate within the elastic range of the material, so the elastic modulus and yield strength are of particular interest. The test coupons will be printed on the same M290 printer, out of the same proposed AL6061-RAM2 aluminum powder that will be used to print the interstage. For our testing, the test coupons will be left with their “as printed” surface finish, and will be tested at low temperature of 0 ℃ and a high temperature of 230 ℃.n Results from these tests are discussed in the results section of this report. Fatigue tests, resonance tests, and surface roughness tests were also conducted.

C.) Error Detection



According to Joe Block and the rest of the additive team, the most common error that has the biggest impact on a 3D print is when the machine has issues with recoating properly. This has a variety of causes such as operators experimenting with the thickness of the powder recoating or the recoating mechanism spreading unevenly. Internal systems in the printer already do have error analysis techniques, but these foci are on thermal imaging that can miss the recoating errors that tend to be the most critical.

The most basic prototype for the error detection system is a demonstration of the capability of computer vision methods to detect a known recoating error. The additive lab saves the images from every print, so to start prototyping the plan was to acquire images from a part that was known to have failed because of a short-feed error. These issues caused by uneven layering of the aluminum powder can usually be spotted by the naked eye, therefore it can be assumed that a computer could be taught to recognize irregular patterns within the printed area of the part. This then could be analyzed compared to the surrounding similar parts to show that the computer could detect this error as well as, if not better than, the human eye.

An ideal short-term prototype is simply demonstrating the ability to “see” errors in an existing part known to be faulty. The long-term prototype would be to run a larger-scale experiment to demonstrate the correlation between computer-found errors and test results found during metallurgic testing.

8. Final Results

A.) Missile Component Design

A1.) CAD Design

Once we decided to pursue a more simple shell design, we then focused on refining the wall thickness and bolt attachments (***Figure 7***). We continually removed material in order to hit our weight specification. The part evolved as we kept testing it using FEA (Finite Element Analysis). Our concept was to reduce the thickness of the internal wall and bolting flanges in order to maximize the strength-to-weight properties.

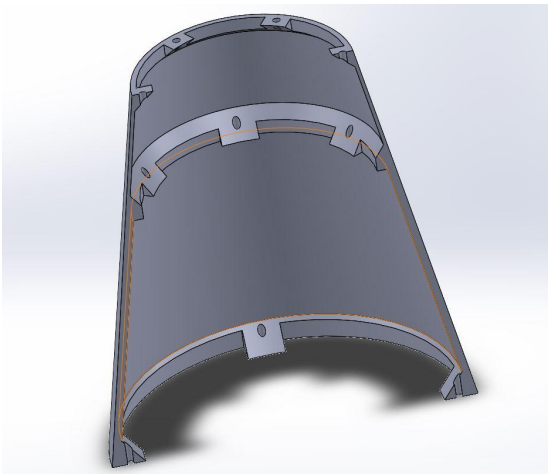


Figure 7: *External shell of the part*

Through multiple iterations, we continuously adjusted the wall thickness, material around each flange, and support angle of each section. As we kept adjusting all these parameters, we

continued the iterative process of reducing mass (***Figure 8***). We also adjusted each structure so that it was easier to print on the M290. The part now then weighed ~3.9 lbs. Our next goals were to further refine the part (using FEA), and concluding which way to successfully mount Mass 3 to the cone without Mass 2 interference.

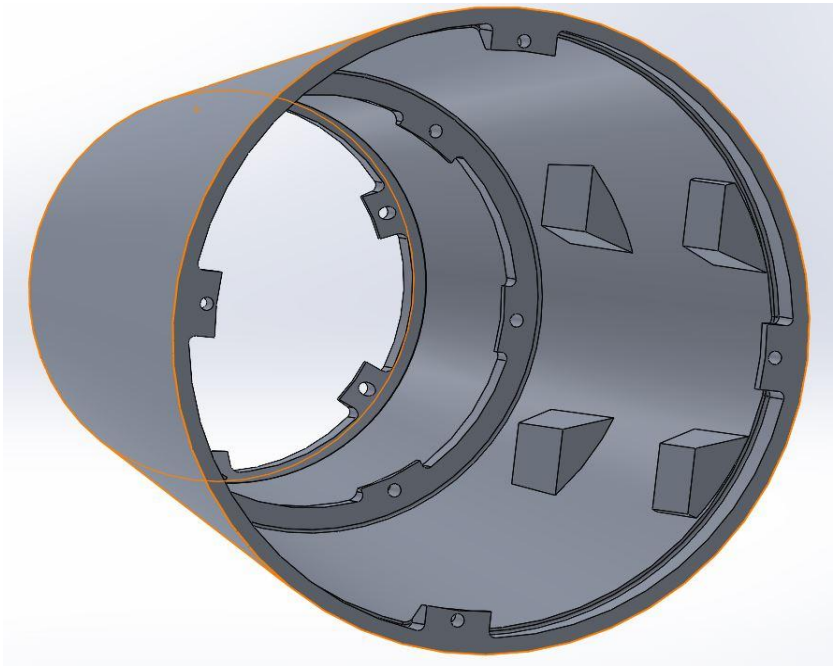


Figure 8: *Thinning shell*

In this analysis, we conducted a stress simulation on the base with the mass 1 load applied. We only did an FEA with mass 1 because we wanted to first initiate the analysis as a preliminary study. Since we were constrained by an elementary familiarity with finite element analysis, we wanted to first verify with other professors and FEA experts to confirm that we conducted the simulation correctly. We also wanted to confirm the forces used in this analysis. Based on the simulation shown in figure 9, the load on mass 1 produced a stress of 96 kPa. This was our first FEA analyzed during the first semester.

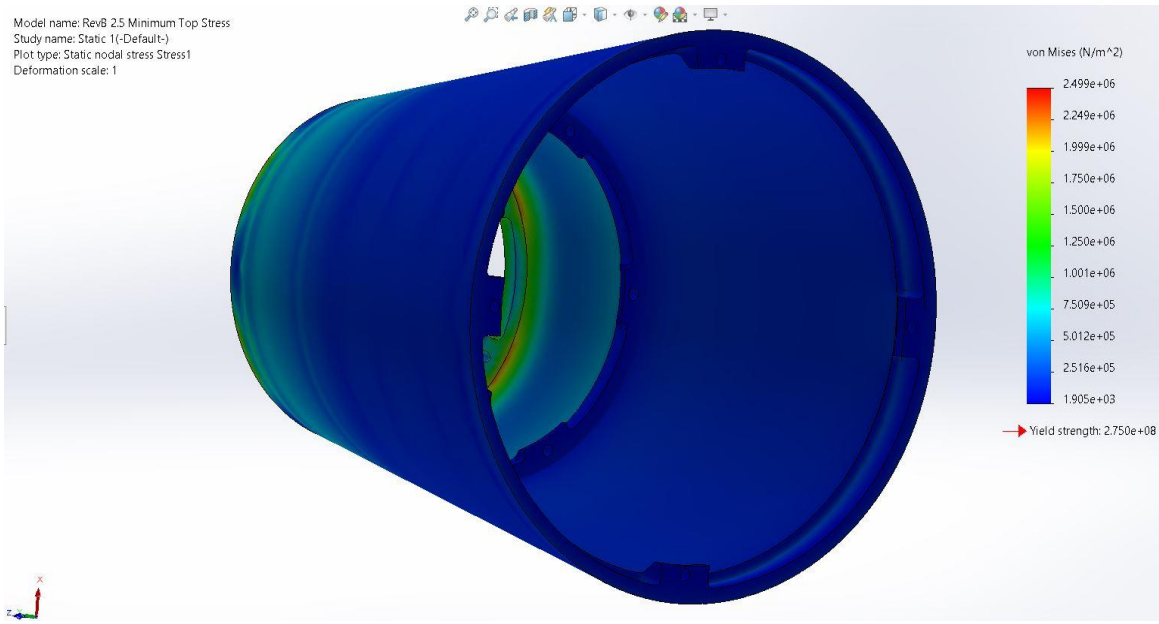


***Figure 9**: Finite Element Analysis of the part*

Another finite element analysis simulation was performed, this time, not only including the weight deduction but also all the three masses that will be mounted into the part. Once this simulation was analyzed (shown in ***figure*** 10*),* with all the three masses being incorporated into the simulation, the maximum stress performed was approximately 6.25 MPa.

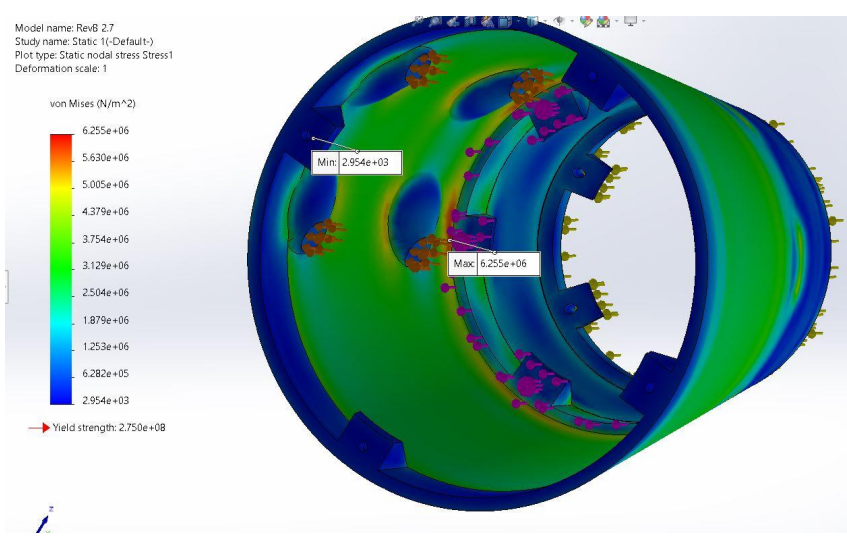


Figure 10: *Second FEA of the part*

Once the initial prototype was printed (***Figure 11***), it was clear that the part lacked strength in the concentrated regions of the flange tabs. In short, we removed too much material, and could see how much stress was concentrated in those areas (***Figure 12***). This presented a dilemma not only in the strength of the part, but also in the production of weak points during the machining process based on the required tolerances, heat, and shear. This was a good example and a consequence of making the part ‘too light’.

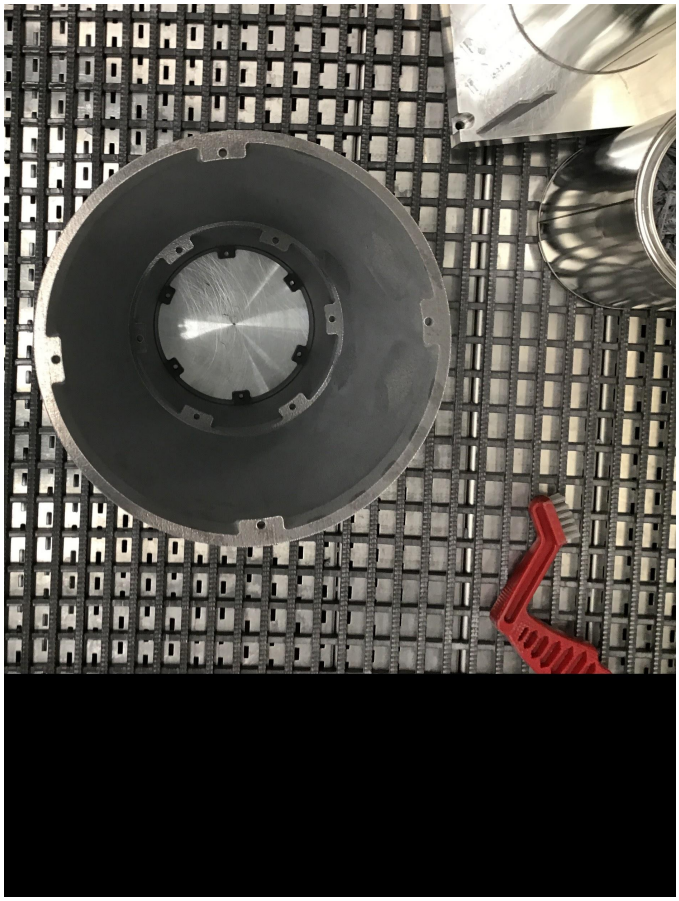


Figure 11: First printed prototype

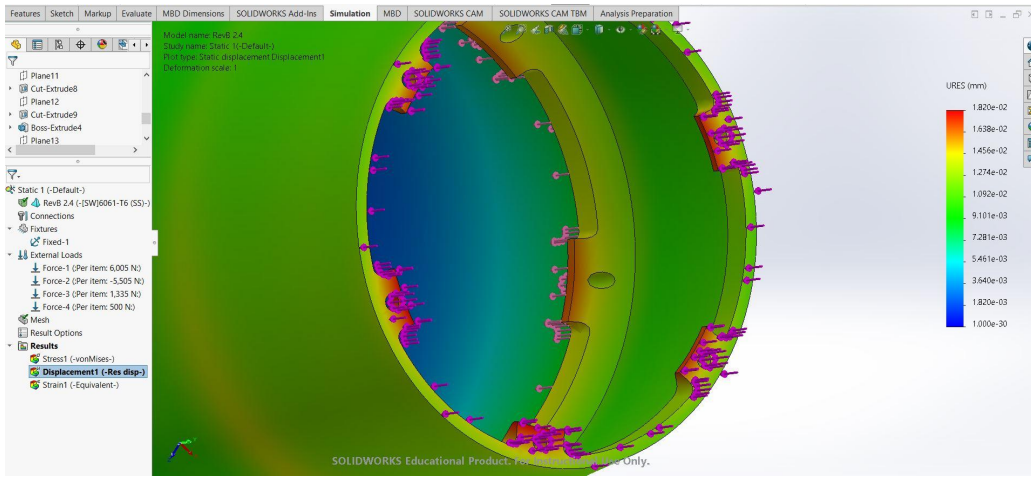


Figure 12: High stress points from FEA

Our next step was to do a quick redesign of the part, adding material to the flange sections where stress concentration was too high (***Figure 13***). That process added about one pound to the part, but it was a much stronger and more robust design. This step allowed the part to be very durable for the machining and post processing stages. Due to restrictions with the M290 printer in this project, our team wasn't able to print and test a second prototype. However, with the feedback of the UCD campus machinist Jac Corless, the new design would have most likely superseded the first iteration's post-processing issues.

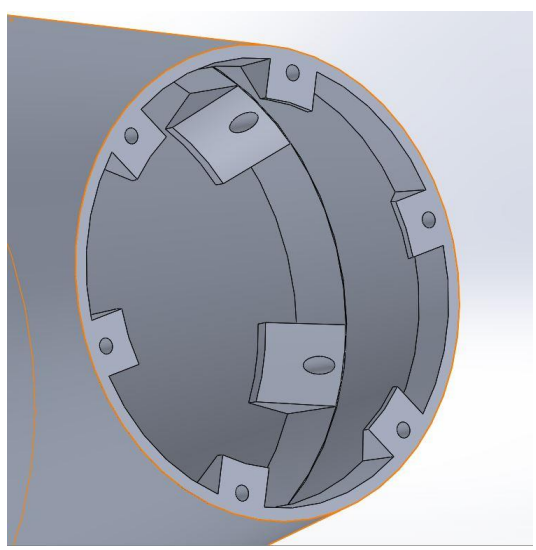


Figure 13: Redesign with added flange material

The final assembly of the part followed. We decided to build simple ‘spacers’ (***Figure 14***) to mount Mass 3 to the internal wall of the cone. The team committed to a rudimentary design for several reasons. One was that it was very easy to print on it’s side with no support material, minimizing costs and maximizing simplicity. Secondly, we could easily thread this spacer to accept any bolt and thread size that we needed during the machining process. Third, if something went wrong, we could simply reprint another example and start the process over again.

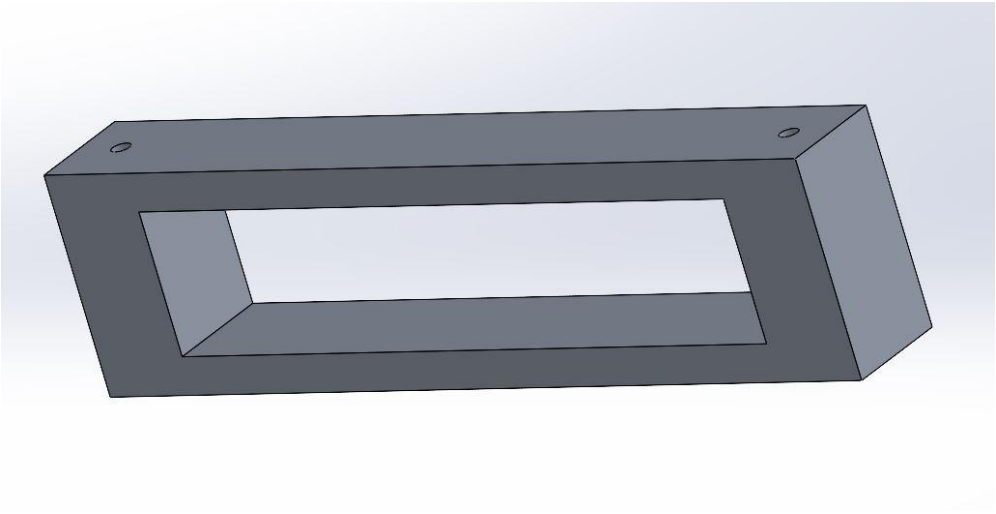


Figure 14: Block spacer for internal mounting of Mass 3

After the decision to design and print the basic spacer, we assembled the part in Solidworks (***Figure 15a,b,c***). Due to time and managerial restrictions, this design has yet to be printed, but we are excited to see the project move forward and a print developed. Overall, and with consultation of various AM and machining experts, the design appears satisfactory and robust to our overall goal weight of less than 5lbs (~4.3lbs total), as well as all other design requirements.

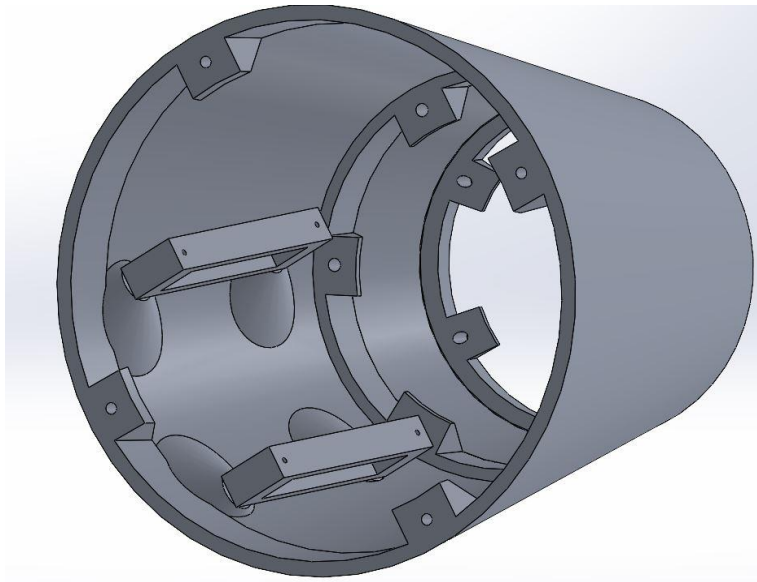


Figure 15a: Final assembly of internal spacer and cone

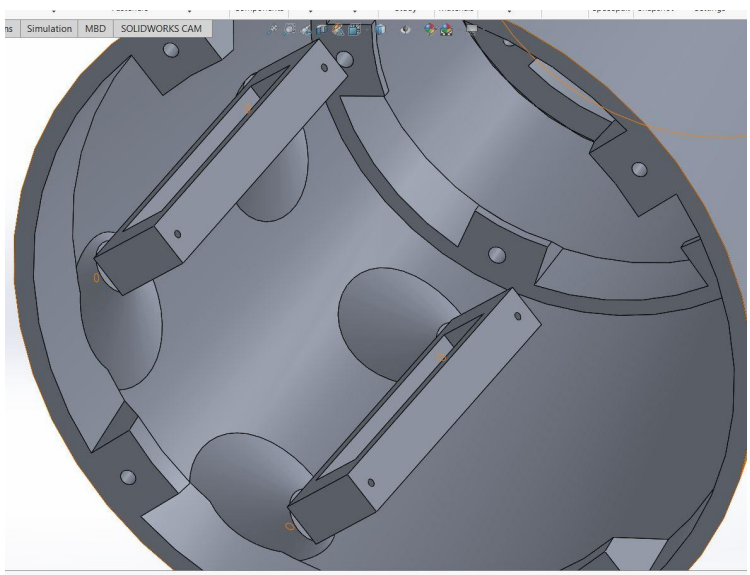


Figure 15b: Closer view of final assembly

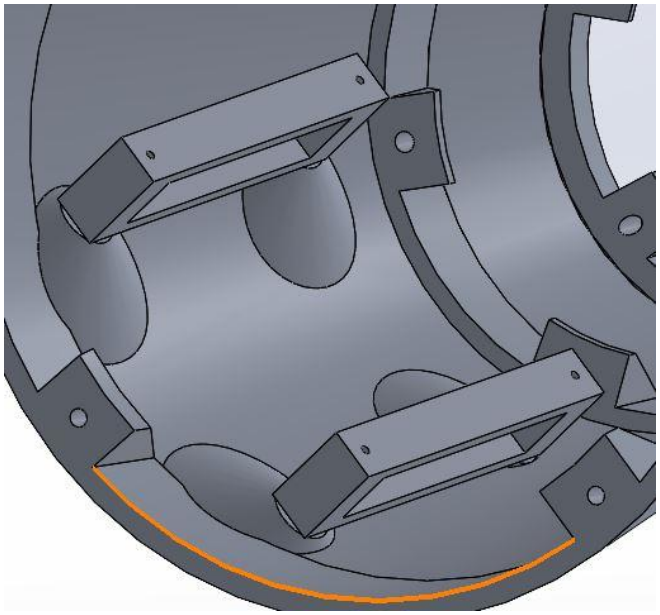


Figure 15c: Another angle of final assembly

A2.) Physical prototyping

The design of the part, once completed, was then prototyped with cheaper material to explore any salient problems with printing or design. With this in mind, a 75% scale PLA version was been printed without error to be visually inspected for defects or suspected printing difficulties. The PLA version allowed us to test the feasibility of proper machining for finishing work.

We consulted Jac Coreless (machinist at the Hub) with the PLA prototype and he confirmed no foreseeable issues with our machining itinerary once the aluminum version is printed, though on closer inspection of the metal print, this was determined to not be the case.

A3.) First Metal Print and Post-Processing

The missile cone was printed successfully on the M290 printer. No support material was needed, and the print had superb integrity. After consulting Mr. Corless in the machine shop, we discovered that too much material was removed on the upper and lower bolting ‘tabs’, causing

an issue with machining. A newer iteration was then devised so that more material was added so that the correct machining could be done without failure of the part.

B.) Material Testing

B1.) Tensile Testing

Tensile testing was conducted on ASTM standard cylindrical dogbone coupons. The coupons were printed on the EOS M290, with AL6061-RAM2 aluminum powder, and exposed to a T6 heat treatment process. Tensile testing was conducted on eight coupons at 150 ºC and four coupons at -20 ºC. Testing was conducted in a thermal chamber with the tensile testing grips inside it. The thermal chamber uses an electric furnace for high temperature testing, and the gas of liquid nitrogen for low temperature testing. Each sample to be tested was placed inside the grips, the chamber was closed, then allowed to come to the proper testing temperature. Once the chamber came to the proper testing temperature, the test temperature was maintained for ten minutes to allow the temperature within the chamber to equalize, and for the sample to reach the testing temperature throughout. For each sample, the ultimate stress, σ_u, yield stress, σ_y, and failure stress, σ_f, was calculated. We intended to determine the elastic modulus, E, of each sample, but were unable to due to unexpected equipment limitations at the extreme high and low testing temperatures. For each sample, the ultimate strength was determined as the maximum amount of stress that the sample was subjected to before beginning plastic deformation. This point can be identified visually as the stress value associated with the highest point on the stress-strain curve, or by using Excel's MAX function applied to the range of stress data associated with each sample. For these tests, Excel's MAX function was used to determine each sample's ultimate strength. Yield strength was calculated using a 0.2% stress offset. A 0.2% offset stress line was plotted with the same slope as the sample's stress-strain curve, extending from the stress-strain curve's y-intercept, up through the

actual stress-strain curve. The yield strength of the material is the applied stress at the point where the offset stress line crosses the stress-strain curve. Failure strength is the applied stress at the point where the sample breaks during testing. This can be observed in the collected data as the point where the applied stress declines abruptly, and the stress-strain curve drops steeply. The average values of these strengths, for the hot and cold tensile tests, can be seen in ***Table 1***.

Test Temperature	Ultimate Strength (σ_u) [MPa]	Yield Strength (σ_y) [MPa]	Failure Strength (σ_f) [MPa]
150 °C	274.2	270.8	191.2
-20 °C	323.1	298.1	286.2

Table 1: *Average High and Low Temperature Tensile Test Results*

These test results from hot and cold tensile testing were used in finite element analysis of the final iteration of the missile interstage. Tensile test results and the associated finite element analysis show that our 3D printed missile interstage will have the strength to withstand the g-forces the interstage will be subjected to throughout the expected range of operating temperatures.

B2.) Surface Roughness Test

_____Surface roughness tests have been completed (great thanks to Dr. Rorrer). The tribology parameters were obtained from our data. These parameters are provided below:

	Smooth	Rough				
Ra	1.09	8.52		Parameters in um=micrometer		
RzD	6.32	41.63		Overall distance=4.8 mm		
Rmax	8.17	48.81				
RzD	7.48	42.18				
RT	8.32	29.99				
Pc	12	7.16				
Sm	0.83	3.53				
Rq	1.36	3.57				

Figure 16a: Surface roughness parameter values

In elementary terms, it can be seen that the various parameters between the smooth and rough coupons vary significantly. Surface roughness parameters R_a, R_z and R_q, which are the most common parameters for describing and are directly proportional to roughness, are several magnitudes higher in the rough vs smooth coupons. R_a, the arithmetic average of the 2D surface roughness profile, is nearly eight times larger in the rougher coupons. This vast difference would be expected to greatly affect fatigue strength (discussed later in this report). R_{max} is the greatest distance (in μm) from the baseline radius.[1] Visually the results can be seen better graphically:

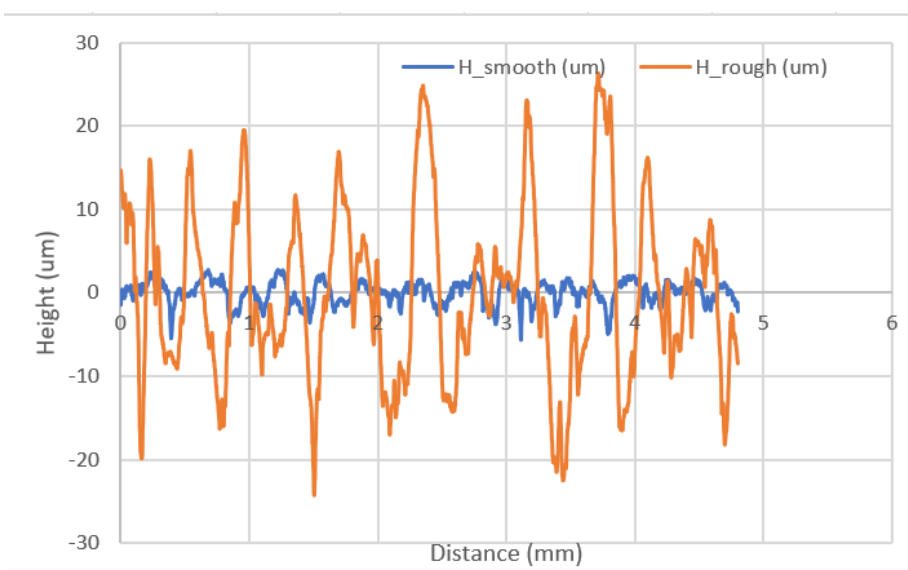


Figure 16b: Surface Roughness graph in terms of varying microheight.

In tribological terms, we consider these differences magnitudinally significant, but without material properties data, such as fatigue strength, we cannot draw definitive performance conclusions from these parameters alone. Therefore, our fatigue testing sought to examine how this difference could affect material properties.

B3.) Fatigue testing

In total, twelve fatigue tests were completed.

Summary:

The goal of testing was to collect and analyze data of the fatigue properties of the A6061-RAM2 provided by Elementum 3D. The samples were heat treated with no surface finish and were as-printed. The goal was to acquire data and compare the rough surface to the smooth machined surface of a post processed coupons, data collected from a previous study by Dr. Yakacki. The test was executed on the MTS machine in CU Denver's Civil Engineering lab. Using the provided yield strength, ultimate strength, and the measured diameter, the maximum and minimum strengths for each test were calculated.

Method:

To attempt a smooth curve The tests started at 90% maximum force to 40% in the 12 samples tested. Using MTS Machine' clamping jaws, the coupons were set and zeroed out on the force scale. The test was set at 10 hz frequency until the force used was below 60%. Then the test was set to 30 hz, which ran until failure. Forces were calculated and set into the

program. The increase of frequency was due to samples expected to exceed 60k cycle with the decrease of force used. The cycles were then recorded at fracture.

		Sy =	285	MPa			
		Su =	315	MPa			
		F =	10,30	hz			
Sample	D = mm^2	Cross Area	%UTS[MPa]	% Force	Max [N]	Min [N]	#Cycles
Z6	6.31	31.27	283.5	90	8865	886.5	12926
Z7	6.31	31.27	267.75	85	8372	837.25	21408
Z8	6.32	31.37	252	80	7905	790.52	26265
Z9	6.33	31.47	236.25	75	7434	743.47	33313
Z10	6.31	31.27	220.5	70	6895	689.5	41805
Z11	6.32	31.37	204.75	65	6402	640.25	52403
Z12	6.31	31.27	189	60	5910	591	70112
Z1	6.33	31.47	173.25	55	5452.17	545.217	168754
Z2	6.33	31.47	173.25	55	5452.17	545.217	64057
Z3	6.32	31.37	157.5	50	4940.77	494.077	78745
Z4	6.32	31.37	141.75	45	4446.69	444.669	128246
Z5	6.33	31.47	126	40	3965.22	396.522	160258

Figure 17. Data collected for fatigue testing showing forces and cycles

Sample	D=m	A m^2	N/m^2	MPa	Cycles
Z6	0.00631	3.13E-05	2.83E+08	283.49	12926
Z7	0.00631	3.13E-05	2.68E+08	267.72	21408
Z8	0.00632	3.14E-05	2.52E+08	251.99	26265
Z9	0.00633	3.15E-05	2.36E+08	236.22	33313
Z10	0.00631	3.13E-05	2.20E+08	220.49	41805
Z11	0.00632	3.14E-05	2.04E+08	204.08	52403
Z12	0.00631	3.13E-05	1.89E+08	188.99	70112
Z1	0.00633	3.15E-05	1.73E+08	173.25	168754
Z2	0.00633	3.15E-05	1.73E+08	173.25	64057
Z3	0.00632	3.14E-05	1.57E+08	157.50	78745
Z4	0.00632	3.14E-05	1.42E+08	141.75	128246
Z5	0.00633	3.15E-05	1.26E+08	126.00	160258

Figure 18: Data collected for fatigue testing showing forces and cycles

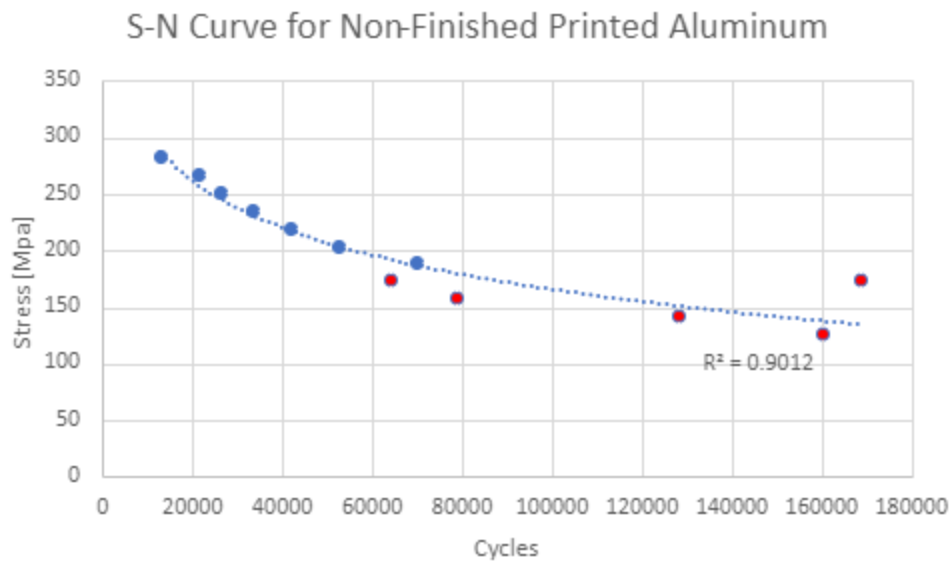


Figure 19: S-N curve for as-printed A6061-RAM2. Fig 3 Curve shows relation between stress and cycles Red being 30 hz blue 10 hz

Conclusion

The data collected and compared to the machine finished coupons was a fraction of the cycles according to the smooth surface data. The test concluded that it fell below the Machine finished coupons but as force decreases it will converge to the 10 M cycle area just as the smooth samples did. Since the smooth surface finish coupons took substantially more cycles before failure than the as-printed surface finish coupons, we were able to conclude that **it is necessary to perform some surface finish machining on printed parts to maximize the structural integrity of the material.**

B4.) Frequency Test

The testing of the part began by using a PCB piezotronics impulse hammer along with an Elvis 2 to collect the data. A sensor was glued to the part using an adhesive. The part was then tested on Different surface faces and one hanging. The hanging was done so that the sample was isolated and the table did not interfere or cause incorrect data. With tests being done the part exceeded the required specification of 20hz and had a natural frequency of 910 hz. After

further research and investigation it was concluded that the data might have been skewed due to where the sensor was glued. These results hint at but do not conclude that the resonance frequency is within required material specifications, and we are excited to suggest additional tests to confirm this hypothesis.

Test 1 Top	Test 2 Hanging	Test 3	Test 4
824.76	1118.68	825.46	823.604
823.45	1115.35	824.448	829.75
294.39	1112.32	825.282	822.354
1001.4	1116.54	824.995	828.281
825.46	1130.66	825.214	826.321
1116	1466.45	825.216	824.446
814.2433333	1176.666667	825.1025	825.7926667

Figure 20: Empirical frequency values.

C.) Error Detection

C1.) Shortfeed Error Detection

The difficulty with shortfeed errors (lack of adequate aluminum powder during the laser pass) is the inability to know that one has occurred until there is a catastrophic failure. Furthermore, the failure often occurs during the printing processes, destroying the print entirely within 10 layers (out of thousands) of the error occurring. This leads to the potential of hidden integrity flaws being present in printed parts and going unnoticed until a part failure occurs.

Fortunately for the additive manufacturing field as a whole, this occurs extremely rarely and often at a small enough scale to not cause any measurable structural integrity issue. The additive team at Lockheed Martin was able to direct us to one notable test coupon that failed material testing most likely due to internal voids caused by shortfeed errors.

For the purposes of this project, we examined 3 material test coupons from print UCD20-29. In ideal re-coatings, the build plate appears to be a perfectly smooth gray surface of aluminum powder. More realistically faint outlines of the previously lasered layer are visible. ***Figure 21*** shows 3 test coupons. Circles have been drawn around them on the right to make it more clear what areas are being examined. The bottom test coupon shows a more standard, acceptable recoating. The middle coupon has some minor defects, but none significant enough to have caused part failure. The top coupon has somewhat obvious white marks that turned out to be significant enough shortfeed errors to cause part failure. Shortfeed errors persist in the top circle for around 10 layers indicating that the persistence of large shortfeed errors like this are the most important to be able to detect.

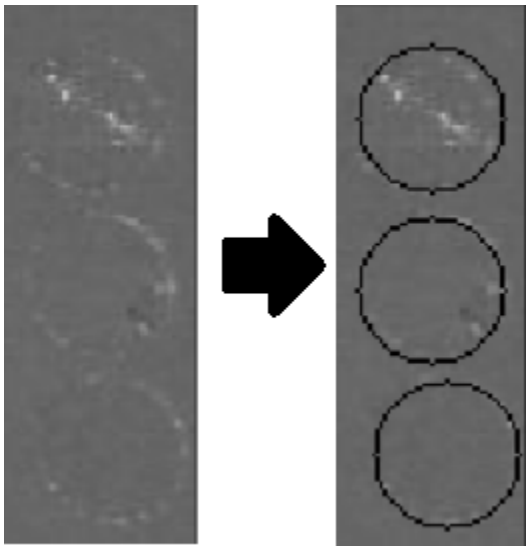


Figure 21: *Layer 1164 of print UCD20-29. Top coupon has major short-feed*

To prove the feasibility of finding print errors such as these, we used python code in the form of Jupyter notebooks to begin basic image processing. The main libraries used were “NumPy” for operations involving arrays and “OpenCV” for general image processing utilities. The code we wrote has 3 main components: finding the region of interest, isolating the errors, and comparing those errors to other layers to find discrepancies.

First, we isolate and crop the desired area. This was done as a manual process to demonstrate the proof of concept. The code crops down the large image to just the 3 desired coupons and circles the area we are looking at. A copy of the image at the same size is also made and circled in order to ignore the circles during the image processing.

Next, we used the canny edge detection method from cv2 to place edges on the image. When this is made more sensitive, it has the effect of highlighting the error regions very clearly in white. This does exactly as desired. Some experimentation using color inversion also showed promise to show errors. This worked extremely well on a very small scale, though some adaptations may need to be made when analyzing a full build plate.

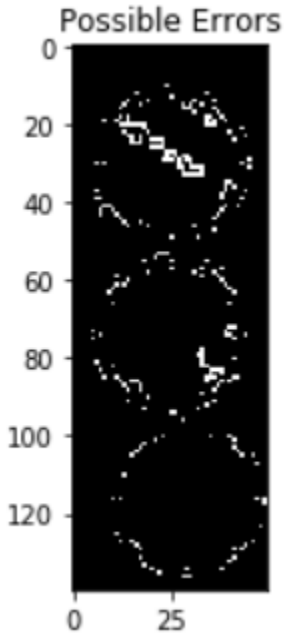


Figure 22: *Layer 1164 of print UCD20-29. Image after sensitive edge detection*

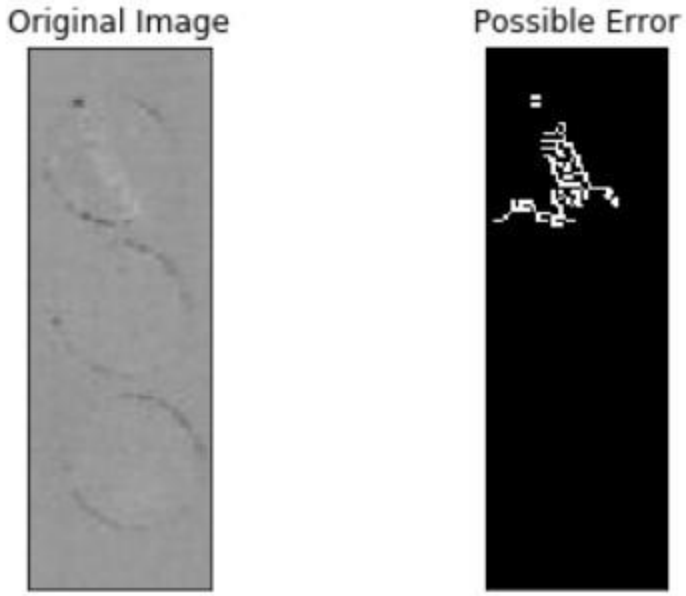


Figure 23: *Layer 1125 of print UCD20-29. Image after sensitive edge detection and color inversion.*

Occasionally the edge detection has the unfortunate side effect of also marking irregularities that are typical like the faint outline of each piece (***Figure 22, middle and bottom circles***). To take care of this, another step was implemented to give a better statistical analysis of “errors” vs regular operation. To get a good idea of the ratio of “errors” to correct operation, a statistical comparison was done on the images after edge detection. The amount of whitespace in the image was compared against the blackspace in order to get numerical values of what could be considered an error according to the image analysis.

Layer	White	Black
1150	164	6836
1151	170	6830
1152	163	6837
1153	201	6799
1154	187	6813
1155	156	6844
1156	161	6839
1157	168	6832
1158	159	6841
1159	159	6841
1160	175	6825
1161	185	6815
1162	235	6765
1163	263	6737
1164	269	6731
1165	210	6790
1166	219	6781
1167	199	6801
1168	190	6810
1169	176	6824
1170	158	6842

Table 3: *Layer Comparison of Whitespace to Blackspace in print UCD20-29*

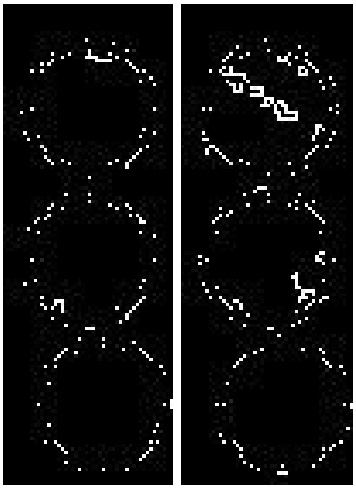


Figure 24: *Print UCD20-29 Error Comparison. Left: Layer 1150 - Right: 1164*

In this test, the numbers show the expected result. In images of layers where there was little or no visible short-feed error, the whitespace numbers were much lower than layers where there was that error. It can also be assumed that this number ratio would be even higher if the image processing was focused solely on the top coupon where the main error took place. The average whitespace jumping from 160 to over 210 in these layers would be noteworthy for the integrity of the part at this location.

Moving forward, now that we know it is possible to link short-feed errors to real values automatically, we worked to improve this script with part detection automation and further ways to make sure the error is detected and confirmed. This will focus on comparing layers of the same piece to show the consistency or inconsistency of the recoating step.

C2.) Part Detection

The next focus of the error detection segment of the project was to map the known part shapes directly onto the build plate. The built-in camera of the M290 is in a fixed location allowing for simpler mapping of pixels to actual locations on the build plate without the need for constant calibration. Using this knowledge, we used OpenCV to apply a perspective warp to change the output image from inside the printer to one of just the build plate.

For the purposes of this demonstration we used print layers close to the midpoint of the build. This was done because the first few layers inherently do not have very clean coatings due to the nature of the M290 printer. These first few layers of poor recoatings have consistently demonstrated to not be consequential as that material is removed while cutting the printed parts off of the build plate.

Figure 25: *Print UCD20-80 After Recoating, Layer 1200. Reshaped to just the build plate*

The indistinct features shown in ***Figure 25*** highlight a near ideal recoating. From this image alone, no amount of image processing could recover the original print data. Fortunately, the same camera takes images from both after the recoating cycle and after the laser pass cycle.

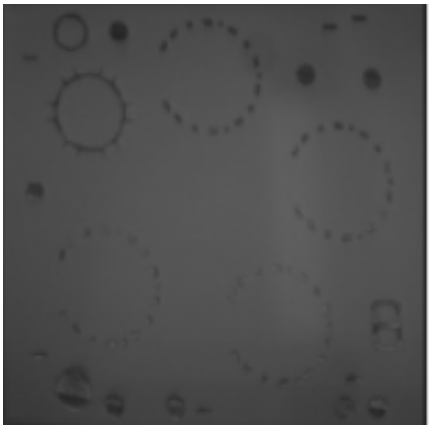


Figure 26: *Print UCD20-80, After Exposure, Layer 1200. Reshaped to just the build plate*

While the features are still not very clear in the after exposure images, (***Figure 26***) they are sufficiently distinct to start to perform various filterings to clean up the image. The goal of cleaning this image is to get it to a form that would be comparable to one that could be produced by slicing the original .stl file. Through experimentation, we came to the conclusion that a mixture of filters including thresholding, median blurring, erosion, and dilation sufficiently

de-noised the image while still maintaining part clarity. It should be noted that this code is specifically tailored to this M290 printer. The camera placement and resolution play a very large role in how the image is filtered. This particular process of sharpening the image also has the downside of having difficulty seeing items that are small enough to be counted as noise. Generally this will produce a clear image for this printer. Similar techniques would have to be explored to apply to other printers with other camera specifications and placements.

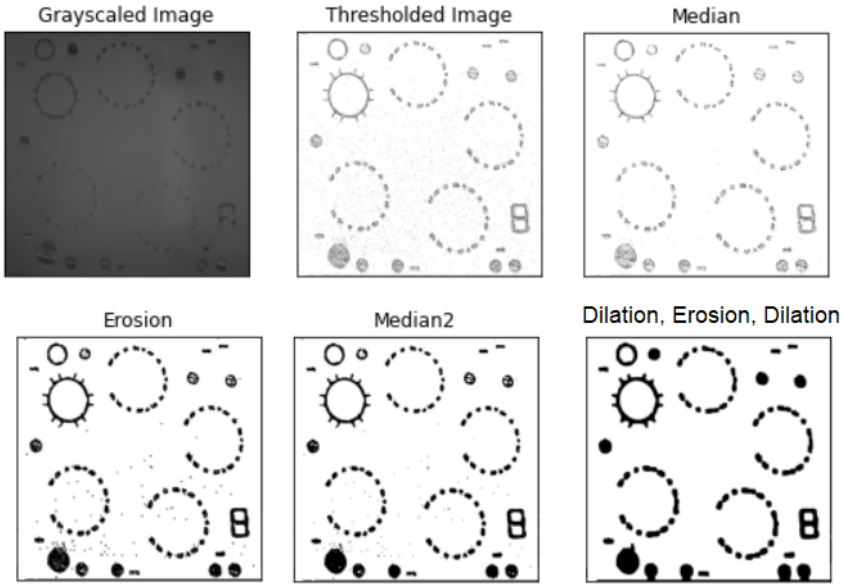


Figure 27: *Print UCD20-80, After Exposure, Layer 1200. Transforming the Image*

We were able to demonstrate that the after-exposure images could indeed be changed into a clear form that is much more reminiscent of a sliced CAD file. (***Figure 27***). The next step was to slice the .stl file and convert it to a file usable by machine vision techniques.

A note for the images and CAD files used below: Our first choice for images and CAD files was our missile interstage design. This would have given us the original .stl file without intellectual property concerns to pair with the after-exposure images. However, at the time our missile interstage piece was printed, the M290 had a software glitch that made it impossible to access the print images and deleted them upon repair of the software. Because of this, this demonstration had to be limited to print UCD20-80 (the print that contained some of the material testing coupons.) For intellectual property reasons we did not receive the CAD file for this print.

The .stl file used in the following images is a mockup made from the after-exposure image of layer 1200. The images shown below are extrusions of 40 micrometers of a real part, not the part itself.

Using the program Slic3r, I took the mockup .stl file for this print and sliced it into 40 micrometer (the size of each layer in the M290). This produced .svg image files for every layer of the print. We took .svg layer 1200 to compare directly to the cleaned up after exposure layer in **Figure 27**. We converted these to .png files using the svglib library in Python. This created a final output file that looks like a slightly cleaner version of the one we created from the after-exposure image. (***Figure 28***) Svglib was not very consistent in converting this image file, so this is the best of the re-running iterations. Full implementation of this method would require optimization of svg to png conversion.

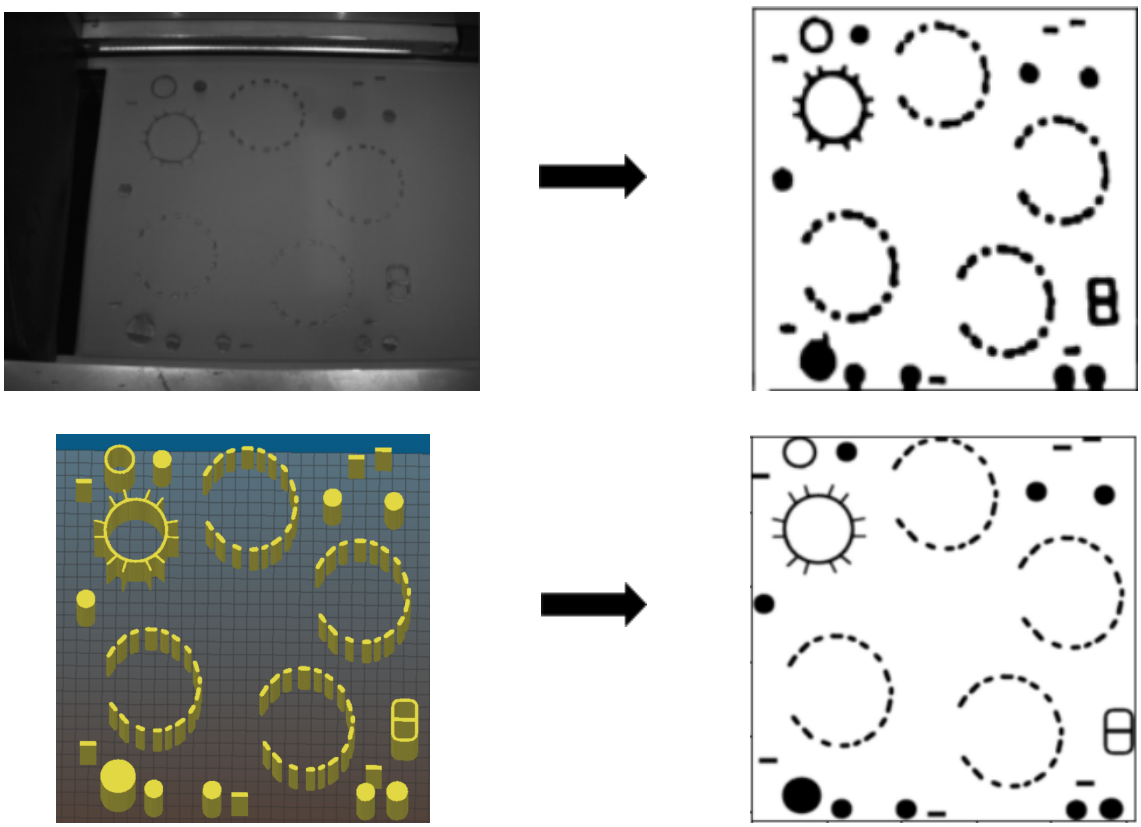


Figure 28: Conversion of a mockup UCD20-80 CAD file

Making these images mirror each other so well allows us to map the location of the CAD file onto the location of the recoating image. Overlaying these images allows us to look directly at what was printed and each layer of the recoating to see which parts were affected. This can eventually allow us to isolate problems in individual test coupons or parts to allow us to disregard or skip tests that we would already know would likely fail. We did not fully implement this exact mapping due time constraints as well as not having the actual CAD file for this print. Our first look at matching via homography proved promising. Additionally, it may be possible to simply warp the CAD file image to match the output image since they should be the same apart from some warp from the angle of the camera. ***Figure 27*** shows how this can be used to see the parts overlaid on the recoated surface.

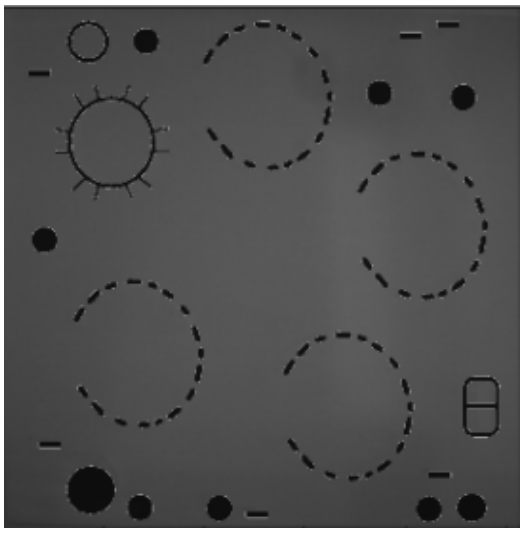


Figure 29: *Sliced .stl file (**Figure 28)** overlaid on Print UCD20-80 (**Figure 25)***

C3.) Intentionally Flawing Prints and Printer Limitations

The main ‘problem’ we encountered in testing shortfeed errors is the reliability of the printer. Shortfeed errors are the type of error that either proves catastrophic or completely inconsequential. We worked with the additive team at Lockheed Martin to determine how to intentionally produce errors that behaved similarly.

We first aimed to see how large an error has to be for it to cause structural integrity failures. This can be simulated by intentionally leaving voids in test prints until density and structural integrity are affected. With the help of the additive team, we were able to determine that the smallest possible feature size the laser pass could avoid was .002 inches (***Figure 30***). This was done by attempting to print voids in a cylinder sized between .001 inches to .01 inches (***Figure 31***). With this knowledge, we printed test coupons with intentional voids in the center with sizes between .002 inches and .01 inches. These have yet to be tested, but the results will prove very useful to know how big an error needs to be before it causes integrity issues.

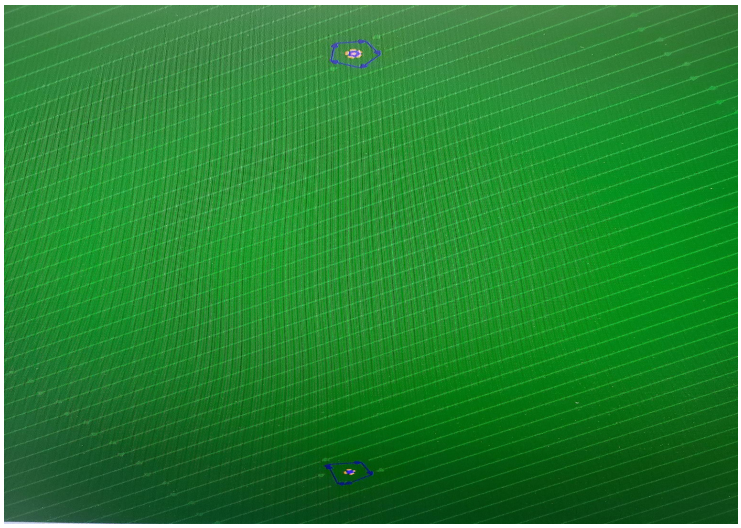


Figure 30: *Laser pass avoiding voids with diameter .002 inches*

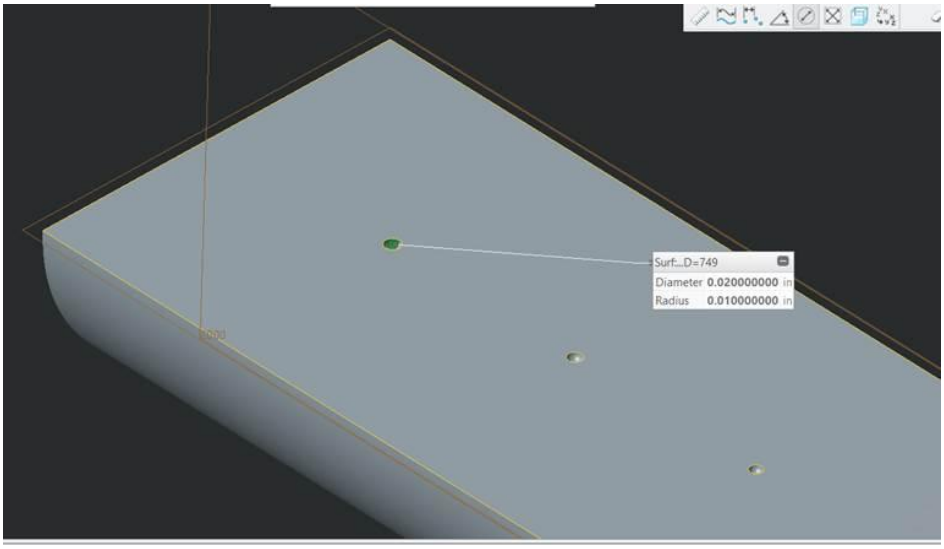


Figure 31: *Cylinder containing various void sizes*

The other concern for being able to detect these issues is the limitations of the built in camera. Using the code we used to map the .stl file to the powder bed, we came to the conclusion that this camera provides a resolution of about .01 inches per pixel in width; about 5x that of the minimum feature size of the printer. This implies that the built-in camera for this printer could only likely detect a recoating error a few times larger than this, perhaps around .05 inches in width. We have yet to determine if errors smaller than this will cause material failure issues. This data provides an excellent baseline for further testing of the limits of error detection in additive manufacturing technology in the future.

An unexplored option for creating shortfeed errors is for the operators to intentionally make the amount of powder spread over a few layers lower than what is required. This would be a final, solid proof of a detection system as test coupons with standard integrity would be printed on one side of the build plate while shortfeeds would occur on the far side of the build plate. This will prove an excellent test of detection systems as this technology progresses.

9. Budget: Cost Analysis, Machine Time, and Turnaround Time

Gantt chart

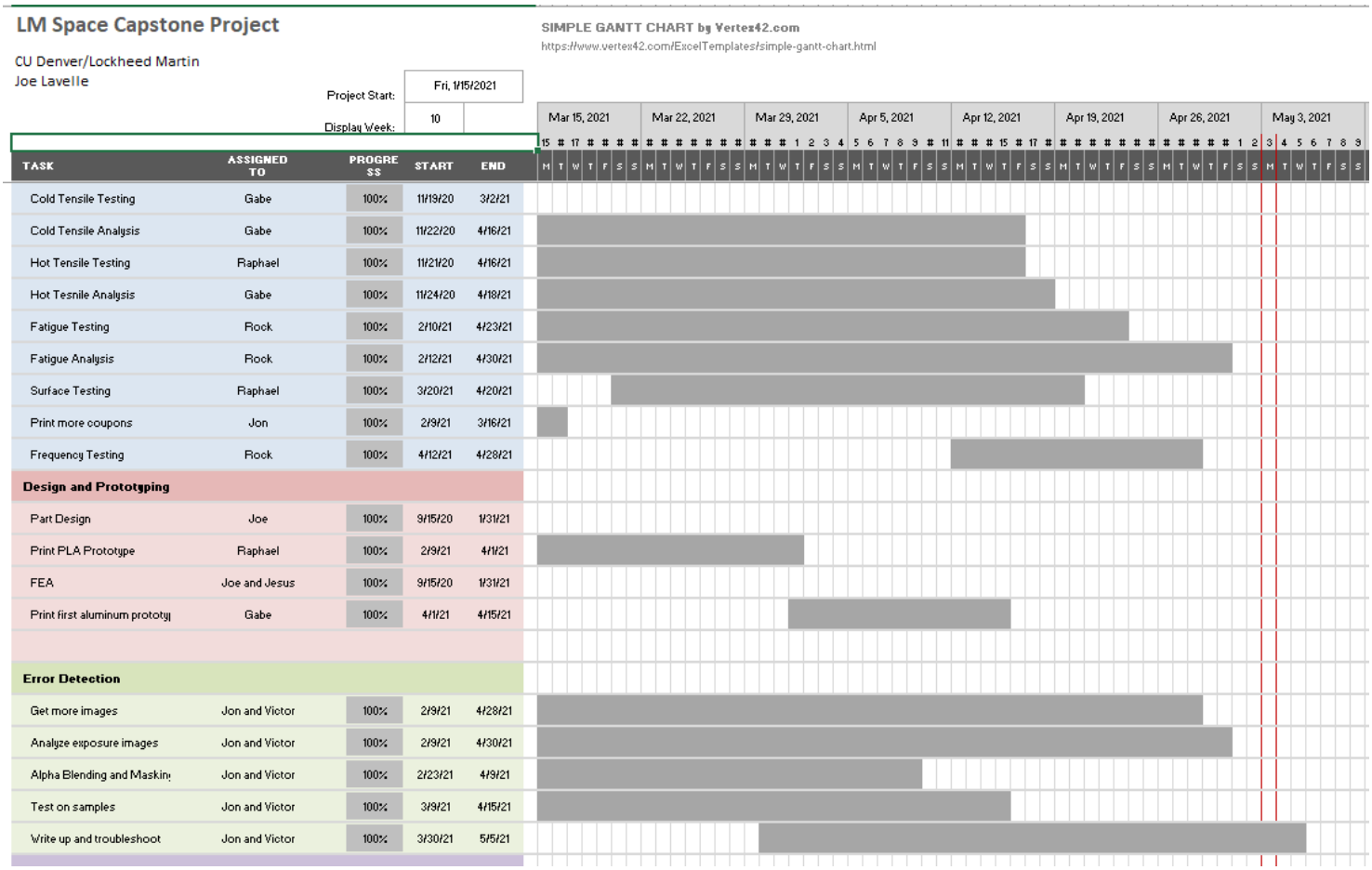


Figure 32: Finalized Gantt chart detailing the completion and timelines of each process in our project.

(Note: The excel sheet providing much more detail on the additive cost calculations is available upon request.)

The cost analysis in this study included a comparison between AM analysis (pertaining directly to this part) and a traditional machining cost analysis (pertaining to a theoretical equivalent part produced through traditional means).

Additive Manufacturing Cost Analysis

A detailed AD cost analysis calculation, one that is used to calculate real Lockheed Martin projects with the M290 Printer, was provided courtesy of Joe Block. With this, we were able to provide a comprehensive cost estimate, one that even includes administrative, testing, and labor costs, to provide our sponsors with an accurate total number of what this part would actually cost. This estimate was fine-tuned as our production plan was finalized. Henceforth is a broad overview of the estimate.

Aluminum powder (A6061-RAM2) currently costs $133/kg. The final weight of the part is about 3.8lbs. Labor costs are generalized at $175/hr, with direct printing labor (printing and finishing) estimated at 15 hours. There is also a cost estimate for the traditional machining that will be needed for this part, provided with the input of Jac Coreless. This estimate involves some uncertainty as we were unable to conduct the machining. However, in a more economic scenario, for a fully-optimized 3D printing design, the through-holes will merely need to be tapped, the cost minimal. In the most extreme scenario, a complex process of externally drilling and filling in will be needed, which will directly cost about $1000. For this initial estimate, we will assume the most expensive scenario. Adding some room for error, this provided us with an initial direct cost estimate for the part of about $4500. Note that the weight of the part is expected to increase, but the machining cost is expected to decrease, altering the final direct cost estimate. Currently, a comprehensive total cost of the part is $26,000. A corresponding figure for the traditional machining cost analysis could not be provided, as the part wasn’t actually produced that way.

Traditional Manufacturing Cost Analysis

A less detailed and rougher estimate for what an equivalent part would cost through traditional (non-3D-printing) means was created with the help of Jac Coreless. Note that the accuracy of this estimate is depreciated as the part in its current design would have to be fundamentally changed to be produced traditionally. A direct cost estimate, with optimal machinery, is in the range of $7000-$9000, however, with the machinery currently available at the Hub (machine shop at CU Denver), the cost would likely increase to as high as $12000. In our analysis as it currently stands, **the part would cost at least $2500 more if made traditionally.** This doesn't take into account the extra time/difficulty that would arise from making the part design compatible for traditional methods. Below provides an estimated range for the AD approach given the current estimate.

M290 Printer:	**$2000 - $6000** (includes required traditional machining)
Traditional Machining	**$7000 - $12000**

Turnaround time

Our sponsors at LM also expressed interest in evaluating the general turnaround time for this part. Joe Block provided us with a rough time estimate of **~2 months**. Turnaround time is very difficult to calculate as it involves many factors, and we were unable to provide a time frame for the traditional-machining comparison. However, we are able to directly compare the estimated machine time, in consultation with Joe Block and Jac Coreless.

M290 Printer:	**25 hours** (10 hours additive labor + 15 hours printing)
Traditional Machining	**80-100 hours**

Note that these figures do not include testing, administration, or any other factors. The total time estimated for the AD approach, including administration/paperwork, is **134 hours**.

Bill of Materials

Below is a table of all accountable expenses. We used school funds (collectively equal to $2200 for the team) to invest on the materials, tools, and the electronics needed for this project. Thanks to generous sponsorship, we were able to remain well within budget.

Table 5. Bill of Materials

Part	Function	Price	Quantity	Retailer	Sub-Total
Ender Series PLUS	prototyping	$580	1	Creality	$580
Testing Coupons	Testing	$40	1		$40
A6061-RAM2 powder (1,569 $/kg)	Printing	4393.2	1	Elementum 3D	4393.2
PLA prototype	prototype	$20	1		$20
heat treatment (sponsored by Dr. Y)	stress relief and part improvement	$500	1		$500
heat treatment (4-4-21)	stress relief and part improvement	$500	1		$500
				Total ($):	**$6,033**
				Total amount sponsored by CU Denver	$1,080
				Total amount sponsored by Lockheed and Dr. Y	$4,953
				ME Budget	$300

				($/person)	
				EE Budget ($/person)	$200
				Total Budget	$2200

10. Patent research

A brief research into potential patents was conducted for our work, particularly with error detection. However, it was ultimately determined that concepts attenuated in this project were too broad to be considered patentable. Given further research into shortfeed error detection, we estimate that it could become patentable when the code produces more quantifiable and tested results.

11. Acknowledgements

Many thanks to Lockheed Martin for sponsoring this project. Our appreciation is especially for our Lockheed contacts Danielle Jacobson and Robert Rauscher. Thank you to Mr. Rauscher for continually providing prompt feedback with suggestions, being very informative and for being available for any questions we had during this process. Thank you to Joe Block for his expertise relating to 3D-printing design, and additive cost analysis. Big appreciation is given to Risheng Zhou, who took many hours out of his busy schedule to help us set up and conduct our tests. Without Mr. Zhou, we would not have been able to collect our data. Thank you to Dr. Yakacki for his very informative suggestions on testing and topology, and his financial support with heat treatment, and thanks to Jac Coreless for input on all machining-related questions. Lastly, a special thanks to Dr. Jensen and Dr. Selman for their protracted extensive support throughout this semester, and thank you to everyone else who has helped us on this project.

12. References

1. Rorrer, R., and Gerdeen, J. Engineering Design with Polymers and Composites. (2nd edition). *CRC Press.*

13. Appendix

All code written is in a jupyter notebook format. While included here, it can be provided at request by Jonathan Malm.

A.) Shortfeed Detection Code

```
# Standard imports
import cv2
import numpy as np
from matplotlib import pyplot as plt
import os

# Read image into array
im=cv2.imread("UCD20_19_1164.jpg")

#Create a blank copy of the array
imblank=0&im
imblank=cv2.bitwise_not(imblank)

#Use matplot to display a copy of the image
plt.imshow(im)
plt.title("UCD20_19_1164.jpg")
plt.show()

#First Crop. Crop copy identically
imcrop= im[200:900, 850:1200]

imblankcrop=imblank[200:900, 850:1200]

plt.imshow(imcrop)
plt.title('UCD20_19_1164.jpg Crop 1')
plt.show()

#Second crop, crop blank identically
imcrop2=imcrop[200:400,150:250]

imblankcrop2=imblankcrop[200:400,150:250]

plt.imshow(imcrop2)
plt.title('UCD20_19_1164.jpg Crop 2')
plt.show()

#Third Crop, crop identically
imcrop3=imcrop2[25:165,35:85]

imblankcrop3=imblankcrop2[25:165,35:85]

plt.imshow(imcrop3)
plt.title('UCD20_19_1164.jpg Crop 3')
plt.show()

#Draw a circle around the area of interest
#Do the same for the blank image
x=cv2.circle(imcrop3, (25,30), 19, 1, 1)
x2=cv2.circle(x, (25,74),19,1,1)
x3=cv2.circle(x2,(29,116),19,1,1)

y=cv2.circle(imblankcrop3, (25,30), 19, 1, 1)
y2=cv2.circle(y, (25,74),19,1,1)
y3=cv2.circle(y2,(29,116),19,1,1)

plt.imshow(x3)
plt.title('UCD20_19_1164.jpg circle 1')
plt.show()

def crop_and_detect(im):

    #read in original image and make a blank copy
    imblank=0&im
    imblank=cv2.bitwise_not(imblank)

    #first crop
    imcrop= im[200:900, 850:1200]
    imblankcrop=imblank[200:900, 850:1200]

    #second crop
    imcrop2=imcrop[200:400,150:250]
    imblankcrop2=imblankcrop[200:400,150:250]

    #third crop
    imcrop3=imcrop2[25:165,35:85]
    imblankcrop3=imblankcrop2[25:165,35:85]

    #circles
    x=cv2.circle(imcrop3, (25,30), 19, 1, 1)
    x2=cv2.circle(x, (25,74),19,1,1)
    x3=cv2.circle(x2,(29,116),19,1,1)

    y=cv2.circle(imblankcrop3, (25,30), 19, 1, 1)
    y2=cv2.circle(y, (25,74),19,1,1)
    y3=cv2.circle(y2,(29,116),19,1,1)

    #return circled picture and blank circled picture
    return x3,y3

file=cv2.imread("UCD20_19_1164.jpg")
area1,area1blank = crop_and_detect(file)

plt.subplot(121),plt.imshow(area1,cmap = 'gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(area1blank,cmap = 'gray')
plt.title('Circles on Blank'), plt.xticks([]), plt.yticks([])

#Pull the directory and reads each into a dictionary
direc='ED1/'

entries = os.listdir(direc)
imdict={}
#print(entries)
image1=cv2.imread(direc+entries[0])

for index,image in enumerate(entries):
    im= cv2.imread(direc+image)
    entries[index]= image.split('_')[1]
    imdict[entries[index]]=im

#Make a list for keys
keys=list(imdict.keys())
print(keys)

#Plot a small copy of the read in images
plotNum=(3,7,1)

for index,key in enumerate(keys):
    plt.subplot(plotNum[0], plotNum[1],
plotNum[2]+index),plt.imshow(imdict[key],cmap='gray')
    plt.title(key), plt.xticks([]), plt.yticks([])

#Crop all of the above images
for key in imdict:
    imdict[key]=crop_and_detect(imdict[key])

#Display all of the images to make sure crop and detect worked
keys=list(imdict.keys())
print(keys)

plotNum=(3,7,1)

for index,key in enumerate(keys):
    plt.subplot(plotNum[0], plotNum[1],
plotNum[2]+index),plt.imshow(imdict[key][0],cmap='gray')
    plt.title(key), plt.xticks([]), plt.yticks([])

keys=list(imdict.keys())
print(keys)

plotNum=(3,7,1)

for index,key in enumerate(keys):
    plt.subplot(plotNum[0], plotNum[1],
plotNum[2]+index),plt.imshow(imdict[key][1],cmap='gray')
    plt.title(key), plt.xticks([]), plt.yticks([])

img=x3
edges = cv2.Canny(img,50,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Detection'), plt.xticks([]), plt.yticks([])


plt.show()

blankimg=y3
blankedges = cv2.Canny(blankimg,50,200)


plt.subplot(121),plt.imshow(blankimg,cmap = 'gray')
plt.title('Original Blank'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(blankedges,cmap = 'gray')
plt.title('Edge Detection'), plt.xticks([]), plt.yticks([])
errors = edges-blankedges

plt.imshow(errors,cmap="gray")
plt.title('Possible Errors')
plt.show()

def cannySubtract(im, imblank):

    edges = cv2.Canny(im,50,200)
    blankedges = cv2.Canny(imblank,50,200)
    errors = edges-blankedges

    return errors


#Manually plot an example

im1150=imdict[keys[0]][0]
im1150edge=cv2.Canny(im1150,50,200)
im1150bl=imdict[keys[0]][1]
im1150bledge=cv2.Canny(im1150bl,50,200)

im1150err=im1150edge-im1150bledge

plt.subplot(131),plt.imshow(im1150,cmap = 'gray')
plt.title('01150 orig'), plt.xticks([]), plt.yticks([])
plt.subplot(132),plt.imshow(im1150edge,cmap = 'gray')
plt.title('01150 edge'), plt.xticks([]), plt.yticks([])

#plt.subplot(143),plt.imshow(im1150bl,cmap = 'gray')
#plt.title('01150 bl orig'), plt.xticks([]), plt.yticks([])
#plt.subplot(143),plt.imshow(im1150bledge,cmap = 'gray')
#plt.title('01150 bl edge'), plt.xticks([]), plt.yticks([])

plt.subplot(133),plt.imshow(im1150err,cmap = 'gray')
plt.title('01150 errors'), plt.xticks([]), plt.yticks([])

im1150err=cannySubtract(imdict[keys[0]][0], imdict[keys[0]][1])
im1151err=cannySubtract(imdict[keys[1]][0], imdict[keys[1]][1])

imdiff=im1150err-im1151err

plt.subplot(131),plt.imshow(im1150err,cmap = 'gray')
plt.title('01150 error'), plt.xticks([]), plt.yticks([])
plt.subplot(132),plt.imshow(im1151err,cmap = 'gray')
plt.title('01151 error'), plt.xticks([]), plt.yticks([])

plt.subplot(133),plt.imshow(imdiff,cmap = 'gray')
plt.title('imdiff'), plt.xticks([]), plt.yticks([])

key1=0
key2=14

im1err=cannySubtract(imdict[keys[key1]][0], imdict[keys[key1]][1])
im2err=cannySubtract(imdict[keys[key2]][0], imdict[keys[key2]][1])

imdiff=im2err-im1err

plt.subplot(131),plt.imshow(im1err,cmap = 'gray')
plt.title('01150 error'), plt.xticks([]), plt.yticks([])
plt.subplot(132),plt.imshow(im2err,cmap = 'gray')
plt.title('01164 error'), plt.xticks([]), plt.yticks([])

plt.subplot(133),plt.imshow(imdiff,cmap = 'gray')
plt.title('Difference'), plt.xticks([]), plt.yticks([])

#print(im1err)


def getBWratio(image):
    white=0
    black=0

    for i in range(len(image)):
        for j in range(len(image[i])):
            if image[i][j]==0:
                black=black+1
            if image[i][j]>0:
                white=white+1
    return white,black


white1,black1=getBWratio(im1err)
print("white=",white1)
print("black=",black1)

white2,black2=getBWratio(im2err)
print("white=",white2)
print("black=",black2)

for key in range(len(keys)):
    imerr=cannySubtract(imdict[keys[key]][0], imdict[keys[key]][1])
    cv2.imwrite(keys[key]+'.jpg',imerr)
    white1,black1=getBWratio(imerr)
    print(keys[key],white1,black1)
```

B.) Part Detection Code

```
# Standard imports
import cv2
import numpy as np
from matplotlib import pyplot as plt
import os
import math
from svglib.svglib import svg2rlg
from reportlab.graphics import renderPM

import tensorflow as tf

#call the rest
physical_devices =
tf.config.list_physical_devices('GPU')
try:
# Disable all GPUS
    tf.config.set_visible_devices([], 'GPU')
    visible_devices =
tf.config.get_visible_devices()
    for device in visible_devices:
        assert device.device_type != 'GPU'
except:
# Invalid device or cannot modify virtual devices
once initialized.
    pass

#important import for this to work
from os import listdir

image_extensions = ['.PNG','.JPG']
testPath='UCD20_80/AR/'

def getImageList(path,imExt=image_extensions):

    #obtain all files of the image extention type
contained in the path
    listfiles = [f for f in listdir(path) if any([ext in
f.upper() for ext in image_extensions])]

    #sort list to match folder contents
    listfiles.sort()

    return listfiles

imageList=getImageList(testPath)
#print(imageList)
print(imageList[0])

#Read Image
testing = cv2.imread(testPath+imageList[12])
#testing =
cv2.imread('AfterExposureExample.jpg')

#Plot Image
plt.imshow(testing)
plt.title('Test Image')
plt.show()

testing2=testing[150:930,200:1150]
print(testing2.shape)
plt.imshow(testing2)
plt.title('Crop Test Image')
plt.show()

def warpSelectImage(image, corners):
    #get the length of the top and bottom of square
    topLen = int(math.hypot(corners[0][0] -
corners[1][0], corners[0][1] - corners[1][1]))
    botLen = int(math.hypot(corners[2][0] -
corners[3][0], corners[2][1] - corners[3][1]))
    #get the larger of the two
    xLen=1028

    #get the length of the left and right of the
square
    leftLen = int(math.hypot(corners[0][0] -
corners[2][0], corners[0][1] - corners[2][1]))
    rightLen = int(math.hypot(corners[1][0] -
corners[3][0], corners[1][1] - corners[3][1]))
    #get the larger of the two
    yLen=1028

    #get the largest of all the sides to be the
uniform size
    sqMax=max(xLen,yLen)

    #determine the size of the desired crop
    newImageSize=[[0,0], [xLen,0], [0,yLen],
[xLen,yLen]]

    #get the perspective matrix and warp to be a
square of the max length

perspectiveMatrix=cv2.getPerspectiveTransform(
np.float32(corners), np.float32(newImageSize))
    newImage = cv2.warpPerspective(image,
perspectiveMatrix, (sqMax, sqMax))

    #return the newly warped image
    return newImage

corners =
np.array([[100,0],[950,0],[0,780],[950,780]])

warpedIm=warpSelectImage(testing2,corners)
plt.imshow(warpedIm)
plt.title('Warped Image')
plt.show()

#grayscaled =
cv2.imread(testPath+imageList[8],cv2.IMREAD_
GRAYSCALE)
grayscaled = cv2.cvtColor(warpedIm,
cv2.COLOR_BGR2GRAY)
threshold =
cv2.adaptiveThreshold(grayscaled,255,cv2.ADA
PTIVE_THRESH_GAUSSIAN_C,\
        cv2.THRESH_BINARY,11,3)

plt.subplot(121),plt.imshow(warpedIm,cmap =
'gray')
plt.title('Grayscaled Image'), plt.xticks([]),
plt.yticks([])
plt.subplot(122),plt.imshow(threshold,cmap =
'gray')
plt.title('Thresholded Image'), plt.xticks([]),
plt.yticks([])

cv2.imwrite("testsave2.png",threshold)

median = cv2.medianBlur(threshold,3)

plt.subplot(121),plt.imshow(threshold,cmap =
'gray')
plt.title('Threshold'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(median,cmap =
'gray')
plt.title('Median'), plt.xticks([]), plt.yticks([])

img = median
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)

plt.subplot(121),plt.imshow(median,cmap =
'gray')
plt.title('Median'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(erosion,cmap =
'gray')
plt.title('Erosion'), plt.xticks([]), plt.yticks([])

median2 = cv2.medianBlur(erosion,7)

plt.subplot(121),plt.imshow(erosion,cmap =
'gray')
plt.title('Erosion'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(median2,cmap =
'gray')
plt.title('Median'), plt.xticks([]), plt.yticks([])

img = median2
kernel = np.ones((5,5),np.uint8)
dilation = cv2.dilate(median2,kernel,iterations =
1)
erosion2 = cv2.erode(dilation,kernel,iterations =
5)
dilation2 = cv2.dilate(erosion2,kernel,iterations =
3)


plt.subplot(121),plt.imshow(median2,cmap =
'gray')
plt.title('Median2'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(dilation2,cmap =
'gray')
plt.title('Image from After Exposure'), plt.xticks([]),
plt.yticks([])

cv2.imwrite("testsave.png",dilation2)

drawing = svg2rlg("testsave2inlayer1.svg")
renderPM.drawToFile(drawing, "tst.png",
fmt="PNG")

testing = cv2.imread('tst.png')

plt.imshow(testing)
plt.title('Crop Test Image')
plt.show()

image = cv2.rotate(testing,
cv2.cv2.ROTATE_180)
image = cv2.flip(image, 1)

plt.imshow(image)
plt.title('Crop Test Image')
plt.show()

cv2.imwrite("stlread.png",image)

resized_image = cv2.resize(image, (1024, 1024))

plt.imshow(resized_image)
plt.title('STL Conversion')
plt.show()
```