Group 6/7: Don Coates, Daniel Hsu, Yang Wang, David Woodruff

Modelling an SLM Printing Production Facility

Abstract

Siemen's Plant Simulation (PlantSim) is a discrete element modeling (DEM) software used to analyze, visualize and optimize production systems and processes, the flow of materials and logistic operations. Recently Professor John Hart's Mechanosynthesis Group developed a PlantSim model to calculate the operational production cost of additively manufactured ("3D printed") parts using a Selective Laser Melting (SLM) machine. Using this model as a basis, we created a simplified version within PlantSim and also created a similar MATLAB model as a comparison. Through our experience, we found the PlantSim learning curve to be great. Customizing many of the pre-defined components within the software took almost as much coding as creating the components from the ground up in MATLAB. The smaller community of PlantSim users and its associated SimTalk 2.0 also made it harder to find help and documentation in problem solving issues for a novice user. However, we did find that once a model of a production line is created, PlantSim is a powerful software that makes comparing different set-ups of that modeled system relatively easy. Hence, it is a very useful tool in helping optimize an existing manufacturing line or finding the optimal set-up when expanding a current production line. In this paper we walk through the process of building the models, compare some results and discuss the advantages and disadvantages of using a DEM software.

I. Introduction and Background

Professor John Hart and members of his Mechanosynthesis Group has been working for the past couple years on building an adaptable and accurate cost model of additively manufacturing parts commercially. So far, much of their effort has been focused at the machine level and making a cost model that incorporates all the applicable costs of 3D printing a part: from the time associated with build preparation to the amortization of the purchase price of the 3D printing machine itself to the myriad post-processing efforts. Based on inputs from one of our team members, Don Coates, and another member of the Mechanosynthesis group, Eldar Shakirov of Skolkovo Instiitute of Science and Technology in Moscow, the group began working on incorporating operational costs of an additive manufacturing production line.

As we have thoroughly studied within this course, Introduction to Manufacturing Systems, a production process does not operate optimally 100% of the time. Bottlenecks occur, parts and processes go idle, inventory builds up, and machines break down. All of these things have associated costs, either in direct monetary costs or lead time of a part. The original model did not take these costs into account and thus would not accurately predict what a part would ultimately cost the manufacturer and eventually the customer. Many manufacturers themselves do not understand these operational costs on a per part basis and instead lump them all into a catch-all overhead cost that get distributed among all their sales in order to recuperate them. By breaking these operational costs down per part, this model will hopefully help both manufacturers better price their goods as well as help customers better understand how much a part should cost them based on realistic operational assumptions.

The step the group took to model these costs was to create a production line simulation on Siemen's PlantSim software. Eldar Shakirov took point on this project and has worked the past few months building a model of a Selective Laser Melting production line for additively produced metal parts with a reference machine of an EOS M100. By entering various parameters into an Excel spreadsheet, the model attempts to calculate both the machine costs and operational costs of producing a batch of parts. The details of this model and the motivation of building it is described in: *Simulating the AM Production Facility: A Configurable Software Tool for Strategic Facility-Level Planning*, a paper Mr. Shakirov recently submitted to the American Society of Mechanical Engineers (ASME) 15th International Manufacturing Science and Engineering Conference being held during the Summer of 2020.

Because of this effort's direct relation to the subject of our class, our group decided to analyze this model to better understand the capabilities and limitations of using discrete element method simulation software such as PlantSim to model a manufacturing process.

II. Project Motivation and Question

Having being introduced to PlantSim during Week 9 of this course, our group wanted to get hands on with the software and compare and contrast its capabilities against how we have been modeling these processes in the class before – via coding in Matlab.

The project included understanding, probing and validating the existing model. In particular, we wanted to understand the model inputs (such as the parameterization of the parts) so we could use them in the creation of own model and maintain comparability between our results and those from Shakirov's original model. We did not re-create Shakirov's model to the same level of detail, particularly as it related to his detailed cost breakdown. Rather, using the techniques learned in class, we developed a MATLAB model capturing a simplified version of the production process. In parallel, we developed a PlantSim model of the same process to compare and contrast. By discussing the model development and methodology with Shakirov, we arrived on a reasonable set of simplifying assumptions that allowed us to proceed with the proposed MATLAB model and still capture the major components of the process. We used Shakirov's original model as a general check that our model results were notionally consistent with those previously derived. However, the bulk of analysis was dedicated towards comparing the MATLAB and PlantSim model development process, the derived results and the flexibility for further development moving forward.

The core question we explored with this project is what are the relative advantages and disadvantages to using an expensive commercial discrete element model simulation software like PlantSim against coding a process within a numerical coding language such as MATLAB? We will share this paper with Shakirov and Professor Hart's Mechanosythesis Group with the hope that it will help inform their future efforts in providing a tool that would be useful to industry in estimating the cost of additively manufacturing a part.

III. Adjustment from Previous Proposal

Our previous proposal was larger in scope that we were hoping to use our simplified PlantSim model in conjunction with our MATLAB code to help Shakirov explore some additional questions and add complexity to the existing model. Specifically, we were hoping to explore some of the following questions:

1. Investigate relationship between production part mix and batching strategies
2. Investigate potential benefits of multiple lines, particularly as it relates optimizing each line for a certain part type (as defined by the parameterizations the model uses to represent a part).
3. Investigate how different part scheduling strategies impact the cost and lead time of parts.
4. Investigate decisions about the line setup when considering this environment as a contract manufacturer for 3D printed parts vs a manufacturing line dedicated to producing a specific assembly or part set. How does demand variability (in terms of part type and quantity) impact the manufacturing environment design.

However, we soon realized that just recreating a simplified version of the model in both PlantSim and MATLAB was large enough scope given the time frame. Getting familiar with PlantSim and its proprietary coding language SimTalk 2.0 was not a trivial effort. Hence, we rescoped our project to an exploration and comparison of the two methods as described in Section II.

We do believe the above questions and resulting answers are of value and hence are included in our discussion of Ideas for Future Work in Section VIII of this paper.

IV. Original Model and Associated Data

The original model obtained from Prof. Hart's research group was an effort to develop the costs associated with a 3D printing manufacturing line given a predetermined mix of parts. The model is developed in the PlantSim environment and takes a detailed approach to breaking production costs down into their constituent parts. Using a predetermined set of parts, the model can explore different batching strategies and production volumes to understand their impact on cost.

The input data for the model is a set of parts parameterized by key variables that impact how the machines interact with the parts through the printing process. Key features of the parts include the part geometry, material, fraction of support material vs part material, required lead time and post processing complexity factor. The set of parts used to analyze the model was developed somewhat arbitrarily based on some common objects. Further inputs to the model include costs of running machines, costs of labor, shift calendars and machine failure rates.

The model initializes by computing a part printing sequence based on the selected batching strategy. Three batching strategies – single part, maximum fill and mixed fill – are experimented with in the model. The total number and type of parts to produce is input via an excel input sheet. In this sense, "demand" is known at the beginning of the printing process.

The model captures a full production line from build prep (downloading 3D print files, batching/scheduling parts), setting up and running the 3D printing machines, heat treatment and cooldown, part separation and post processing. The model includes workers working on 9 hour shifts, periodic maintenance schedules for the machines, machine failure rated where applicable and detailed costs assignments to the steps in the process. The paper published by Prof Hart's group explaining the existing model approach is included with this submission for reference. We develop our description of the modelling approach in the next section and identify areas where we provide simplifying alternatives to the original model in the construction of our own.

V. Modelling Approach

Taking into account that some of the concepts in Shakirov's model are difficult to implement in MATLAB, we decided to adopt a simplified model for the SLM printing process. In particular, the simplified model focuses on investigating how different operation strategies and machine parameters impact the printing process within a 3D printing shop from the perspective of lead time but not cost, as MATLAB is hard to incorporate all complexity regarding to cost terms, such as operators labor cost.

Figure 1 illustrate the concept of modelling the SLM printing process within a shop, with each rectangle representing a machine and each triangle a buffer. Our MATLAB code implements backward tracking strategy, i.e. at each time step, the state and action of the current machine or buffer depend on the state of previous machine or buffer.

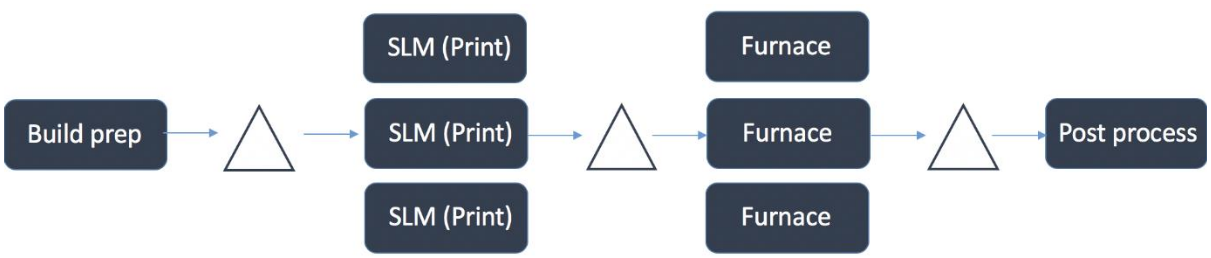

***Figure 1**: Simplified Process Model*

Build Preparation

The first step of the SLM printing process is build preparation, which is carried out by a computer program where print supports will be generated and the parts will be consolidated into batches following predefined batching strategy and order sequence. The prepared batch will stack up in the buffer, waiting to be printed in the next available SLM printer following a first in First Out (FIFO) policy. A constant build preparation time is adopted for each build job, i.e. each batch, regardless of batching strategy.

SLM printer

Different batching strategies are adopted for different purposes. The simplest batching strategy is *single fill* that assigns a single part to each build job, i.e. each single part presents a batch. In our case, we used a more complex strategy called *mixed max fill* to create the batch by allocating the maximum number of parts per batch constrained by the effective build area of the printer, which aims to maximize the space occupancy or fill rate of the printer. Therefore, the printer will be filled with different part types to reach the highest possible fill rate. Key assumptions for the printer include:

a) One to three identical printer(s) with a constant print rate per batch volume, effective build area, failure rate and repair rate
b) Normally distributed setup, recovery and powder refill time, which apply for each build job

Furnace

The printed batches will then be stored in the buffer that is shared for all printers. Following FIFO policy, printed batches will be loaded in the furnace and will be cured for a constant period. The number of furnaces corresponds that of printers, although batches can be loaded to each furnace, regardless of in which printer they were printed. Similar to printer, furnace also follows constant failure and repair rate.

Post Processing

Heat-treated batches will then pile up in the buffer before going to the post processing step, which consists of bandsawing and exterior finishing. While the bandsaw rate is a constant per batch area that need to be cut from the build platform, the remaining time spent on exterior finishing is proportional to individual part volume. There is one bandsaw station and one exterior finishing station in sequence, regardless of number of printers or furnaces. It is also worth mentioning that all jobs throughout the SLM printing process are considered batches, except for jobs at exterior finishing where batches disintegrate into parts that are processed individually, as parts are cut from batch build platform at the previous bandsaw station.

VI. Empirical Findings – MATLAB

1. Throughput performance of model

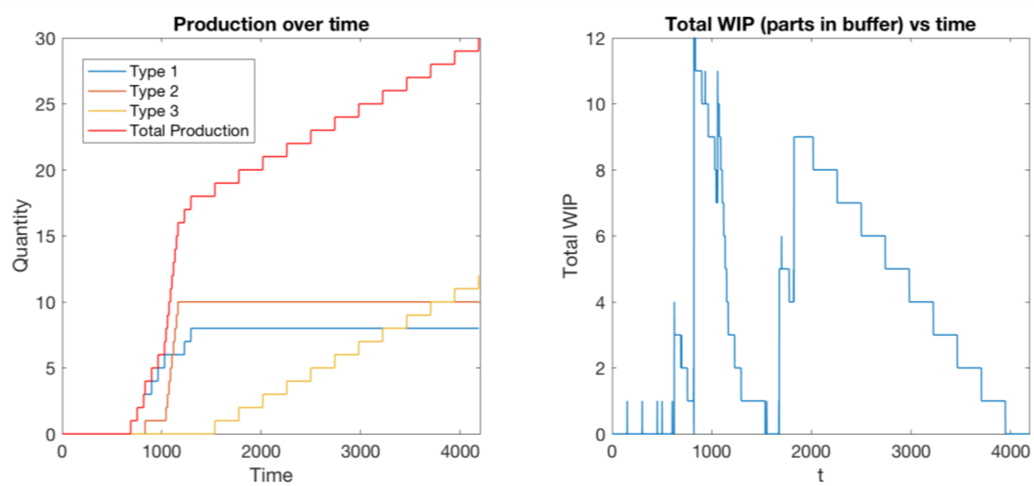

Figure 2: *(left) production of each part over time (right) total number of parts in all buffers over time*

The following batching strategy was created for the purpose of analyzing model throughput and comparing across models. The batching sequence for production is defined as follows:

- Batch 1: 1,1,1,2,2,2,2,2,2,2,2,2
- Batch 2: 1,1,1,2
- Batch 3: 1,1,3,3
- Batch 4: 3,3,3,3,3

- Batch 5: 3,3,3,3,3

The batching sequence was selected because it represents the mixed maximum fill strategy for eight (8) part type 1, ten (10) part type 2 and twelve (12) part type 3 based on the lead time priority assigned to each part assigned in the input file. The quantity of each part and number of part types was selected arbitrarily from the available part parametrizations. Shakirov's model computes this batching strategy directly so our group arranged the same input sequence for our MATLAB and PlantSim models for comparative analysis.

We see from the production plots that the progress of each part is consistent with the order in which they were sequenced. Because the printing is proportional to part mass and generally the most time consuming component of the process, it expected that we see the production rate of Part 3 appear the slowest in the figure.

2. Average lead time and proportion of time spent in buffer decrease with more machines

After several runs of the MATLAB model under different scenarios, we found that the bottleneck of the process can change depending on the model setup. For example, in a one machine (one SLM printer and furnace), we notice that the batches tend to accumulate upstream of the SLM printer. This is supported by the buffer quantities and overage time in the buffer illustrated in the figure below.

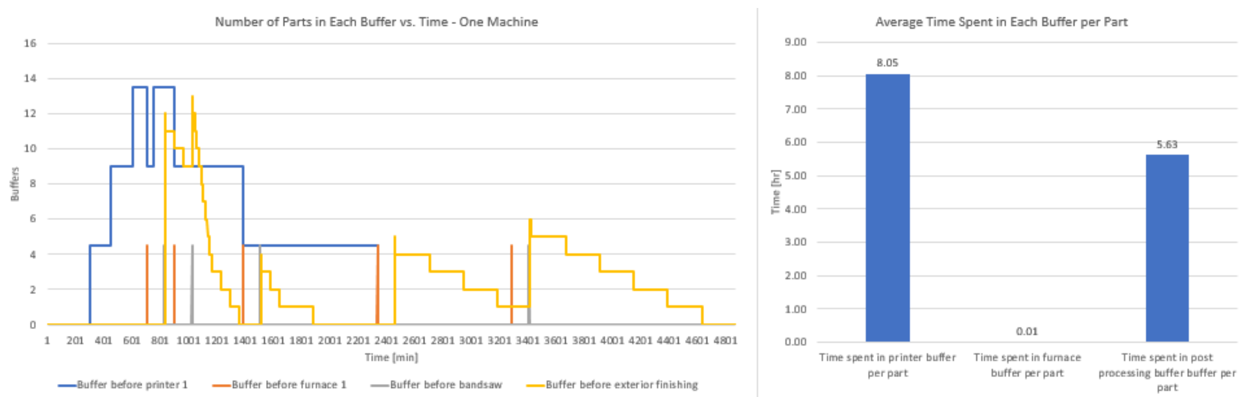

Figure 3: (Left) Buffer contents as a function of simulation time. (Right) Average part time spent in each buffer.

As a result, we expect that adding more SLMs to the process will increase throughput and potentially shift the bottleneck process of the line. The model was modified and run for scenarios *One Machine, Two Machine* and *Three Machine,* where the numbers of printers and furnaces are both reduced to one and two from three, respectively.

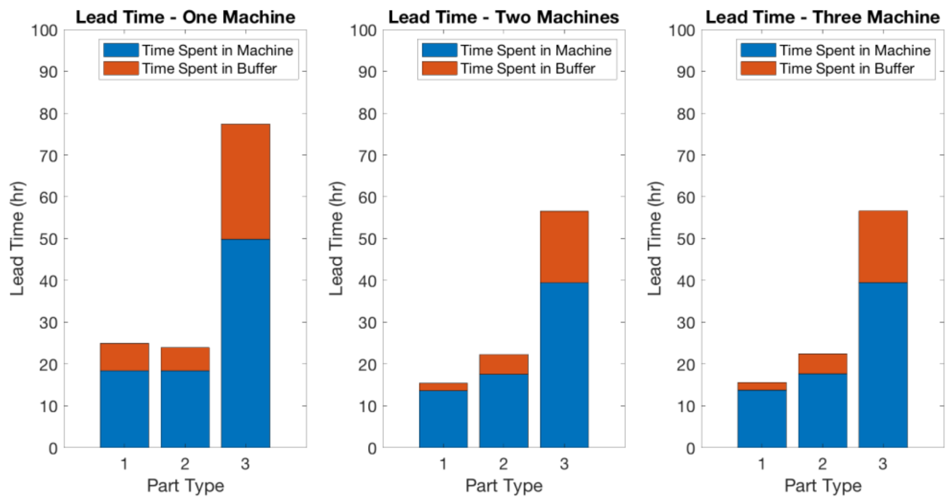

Figure 4: *Lead time vs. Part Type for one-, two- and three-machine scenario*

As shown in Figure 4, as the number of machine increases, lead times of all three part types decreases. While there is a clear drop in lead time from One Machine scenario to Two Machine scenario, no significant decrease is observed from Two Machine scenario to Three Machine scenario. The reason behind it is for the studies case with given part and machine geometries as well as order sequence and batch sizes, more machines than two only bring redundant production capacity. Another interesting finding from the figure is that while the decrease in Time Spend in Buffer is intuitive as more machines bring more production capacity, the decrease in Time Spend in Machine can be interpreted by the fact that the redundant machine filled in the capacity gap that happens when one machine is down according to predefined failure and repair rate.

From this analysis, we also observe in the three-machine, three-furnace scenario, the post processing station is the bottle neck of the simulated SLM process, as only one bandsaw and one exterior finishing station are available. In particular, the post processing station is critical to the lead time of part types with high priority. Since we adopted the FIFO buffer policy, parts with low priorities are grouped into batches with low fill rates and subsequently printed faster, causing them to be post processed by limited resource first, resulting in unexpected long wait time in the buffer of parts with high priority. To address this problem, we adjusted the buffer policy after furnace. Instead of FIFO, the new policy allows the post processing station to identify the batch with high priority part(s) from the piled buffer, and subsequently pick up and process that batch first, leading to shorter lead time of high priority parts.

A similar analysis was completed running Shakirov's original model for comparison. The corresponding results are indicated below

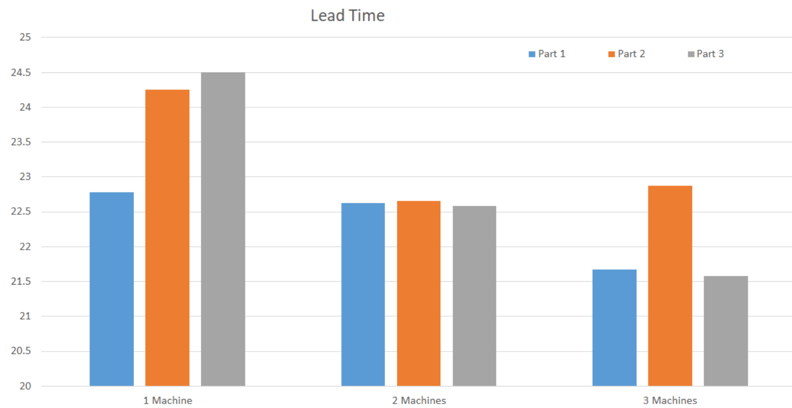

Figure 5: Lead time vs. Part Type for one-, two- and three-machine scenario in PlantSim

The same, obvious general trend of lower lead times by adding machines was shown. However, the specific distribution of lead times between the parts is obviously different. The reason behind this difference is due to the respective batching strategies that were not quite the same between the two models.

3. Average lead time can be reduced by choosing a lower batch size

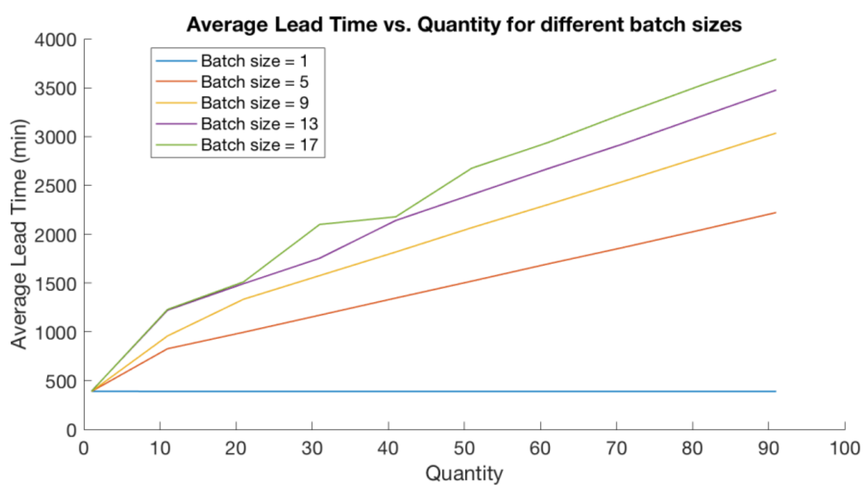

Figure 6: *Average lead time against quantity of parts produced for various batch sizes*

Choosing the right batch size is important because it directly affects the flexibility and lead time of the production. Figure 6, shows a plot of average lead time of a single part for different batch sizes at different production volumes. From the onset we can see that average lead time increases as production volume increases, with the exception when batch size equals 1, which results in constant average lead time regardless of how many parts are produced. This observation matches with our intuition: a batch with only one part will finish printing sooner and move along the production line faster than a larger batch that requires longer printing time. The larger the production volume, the more time is saved by having a smaller batch size. While this holds true at every batch size, the effect becomes less pronounced at higher batch sizes. The larger the batch size, the less of an impact does changing it have on the average lead time.

What this means in a practical sense is that average lead time can significantly be reduced by reducing batch size. However, this reduction in lead time may not be as prominent as what was shown in Figure 6 if one considers the setup time which may be constant to each build.

4. Throughput performance of model

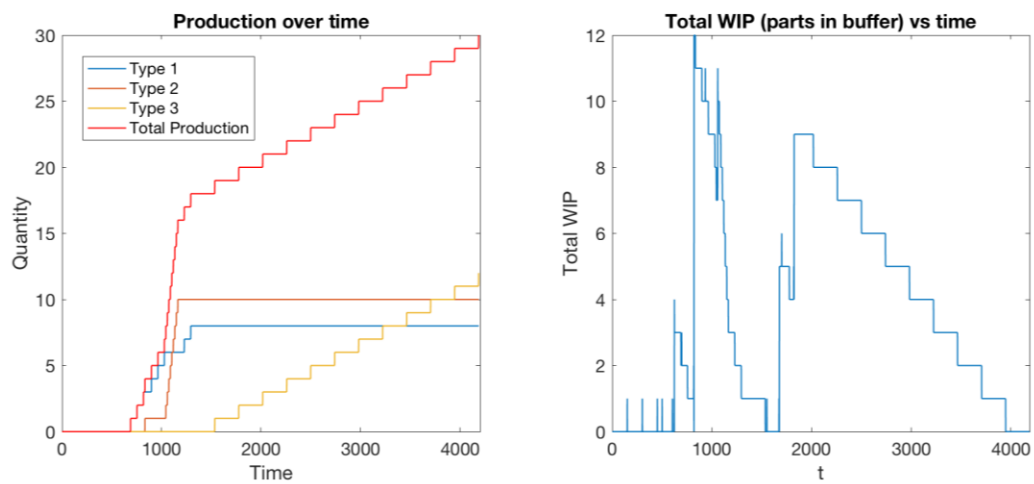

Figure 7: *(left) production of each part over time (right) total number of parts in all buffers over time*

The following batching strategy was created for the purpose of analyzing model throughput and comparing across models. The batching sequence for production is defined as follows:

- Batch 1: 1,1,1,2,2,2,2,2,2,2,2,2,2
- Batch 2: 1,1,1,2
- Batch 3: 1,1,3,3
- Batch 4: 3,3,3,3,3
- Batch 5: 3,3,3,3,3

The batching sequence was selected because it represents the mixed maximum fill strategy for eight (8) part type 1, ten (10) part type 2 and twelve (12) part type 3 based on the lead time priority assigned to each part assigned in the input file. Shakirov's model computes this batching strategy directly so our group arranged the same input sequence to our MATLAB and PlantSim models for comparison.

We see from the production plots that the progress of each part is consistent with the order in which they were sequenced. Because the printing is proportional to part mass and generally the most time consuming component of the process, it expected that we see the production rate of Part 3 appear the slowest in the figure.

VII. Empirical Findings - PlantSim

In parallel with our development of the Matlab model, we created a simplified version of the PlantSim model created by Prof. Hart's group. The intention was to create a set of results more directly comparable with the simplified assumption set of our Matlab model. When we originally obtain the model from Prof. Hart's group, we were unable to run it due to a routine in the model which relied on a functionality within PlantSim that our software license did not allow for use. We were able to work with Shakirov to overcome this difficulty and eventually obtain results from the Shakirov's model as well as our own simplified version.

The PlantSim model we developed closely followed the structure of the Matlab model described earlier. Our model did not attempt to closely track costs as per Shakirov's model, but rather was focused on throughput of the line and lead time. We simplified the model by pulling out the automatic batching code from the PlantSim environment and feeding in directly the intended part sequencing for the batches. This facilitate comparison across all models. The results obtained below follow the same batching strategy as described earlier. We introduced a failure rate to the SLM machine that is not present in Shakirov's model.

In a similar manner to Shakirov's model, we use the input parts sequence to generate batches which are housed in PlantSim container objects. These are formed at the build prep station when the model is initialized. The batches move through the process before getting split apart after the first post processing stage (representing the bandsaw). This was one of the more challenging components of the model to implement.

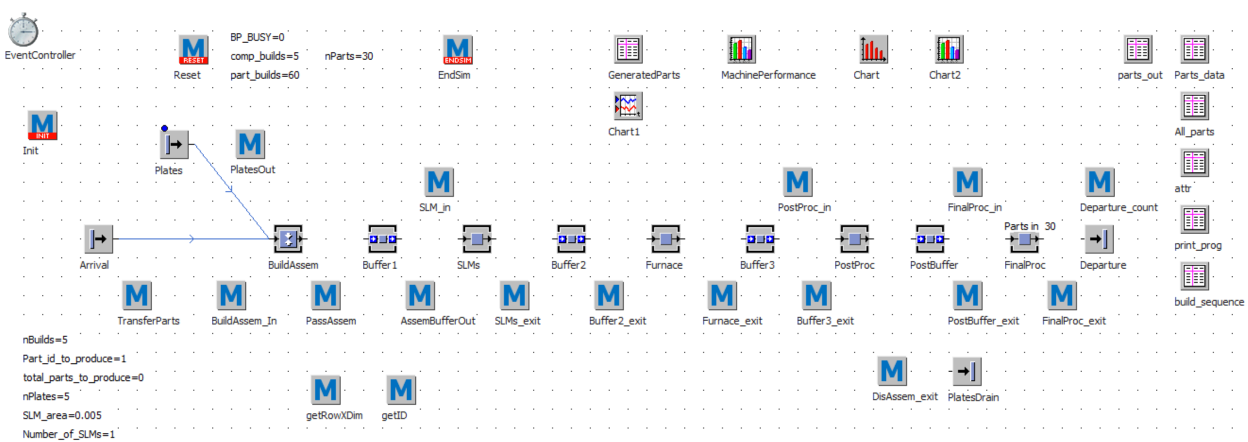

Figure 8*: Visual representation of the simplified PlantSim model developed to replicate the printing process.*

Following a similar construction to the Matlab model, the simplified PlantSim model contains the following machines – Build Assembly (BuildAssem), SLM (single machine only), Furnace, Bandsaw (represented by PostProc) and Final post processing (represented by FinalProc). Buffers are located between each step as follows; Buffer 1 between BuildPrep and SLMs, Buffer 2 between SLMs and Furnace, Buffer 3 between Furnace and PostProc (Bandsaw), PostBuffer between PostProc and final processing (FinalProc).

We obtain the following results running our model along side Shakirov's PlantSim model for the batching strategy described in the previous section.

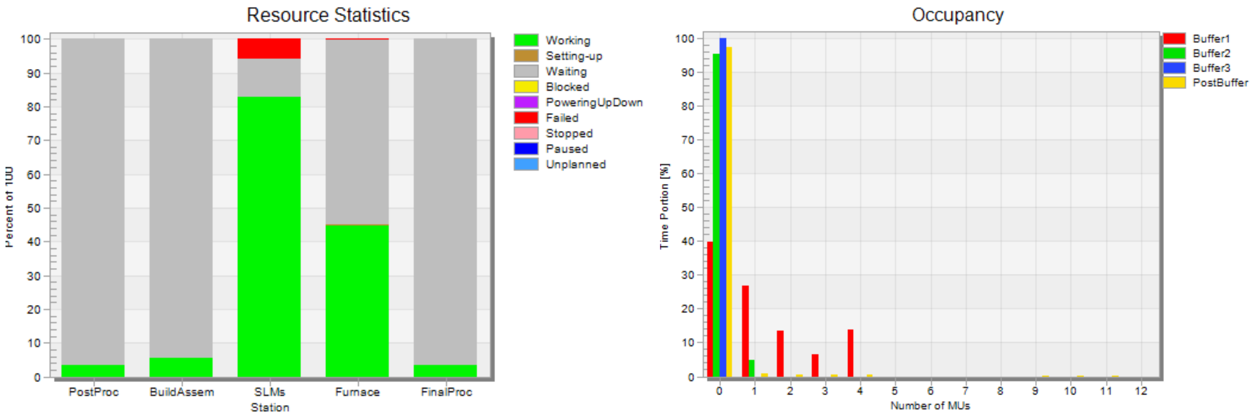

***Figure 9**: PlantSim results for the simplified model. (Left) Percentage time spent in various states for each machine in the process. (Right) Distribution of time each buffer spent with quantities of parts or batches in the buffer (the relevant quantity applies).*

As a means of comparison, the same output is produced from Shakirov's model. We note that because there is not buffer prior to the SLM machines in Shakirov's model, we see that the BuildPrep station is "blocked" when there aren't SLMs available to process a batch. This differs from our model where a buffer is located between the BuildPrep and SLMs (Buffer 1).

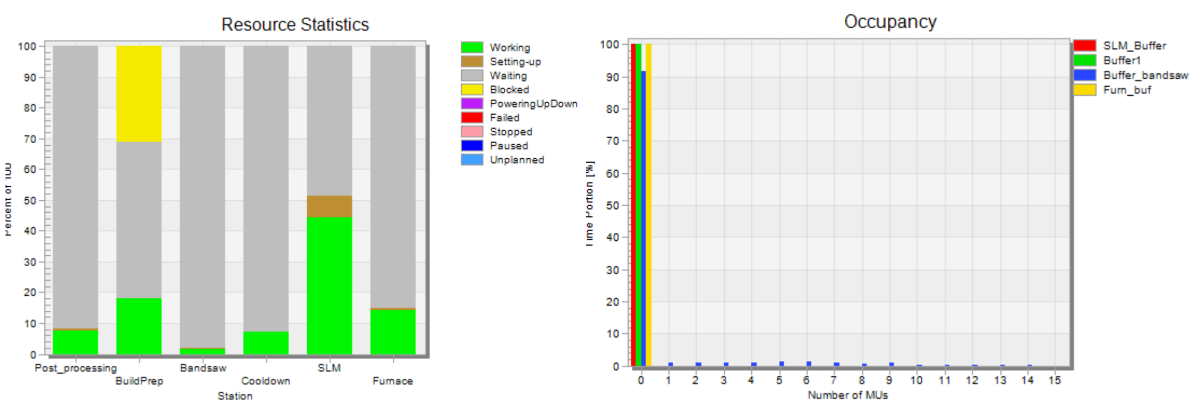

***Figure 10:** PlantSim results for Shakirov's model with the selected batching scheme. (Left) Percentage time spent in various states for each machine in the process. (Right) Distribution of time each buffer spent with quantities of parts or batches in the buffer (the relevant quantity applies).*

The models produce very similar results after accounting for some of the differences that obscure direct comparison. For instance, we see that a large proportion of the BuildPrep time is spent "Blocked". In our model, we have a buffer following our Build Assembly stage. We see that this buffer spends about 60% of its time with non-zero contents meaning that the batch processing of the SLMs is on-going and the printer is unable to take another batch. We also observe that the relative percentages of simulation time each machine spends working vs waiting are similar for the two models with SLMs having the highest percentage in each case. The failure mode of the SLM machine in our model is evident and reflects the 95% machine availability used to model the machine. This was somewhat arbitrarily chosen to prove the functionality in the model but not something that was included in Shakirov's model. In both cases, we see the final buffer – Post Proc buffer and Furnace Buffer – reach 12 units for a very short period of time which corresponds to when the first batch is broken up into individual parts after the bandsaw. Lastly, Shakirov's model has a greater overall simulation time to process the parts, generally falling around 1.5

days to complete the work where ours is closer to 1 day on average. This appears to be due in part to the assembly stopping during blockage rather than being able to continue and supply the buffer.

VIII. Comparison of Modelling Techniques

In this project, a primary objective was to tackle a modelling problem using different techniques to compare the process of developing the model and the consistency of the results to inform how we might approach a similar problem in the future. We essentially had three models – a Matlab model developed from scratch (using the class lab code as a basic structure), an existing PlantSim model that was provided to us from Prof. Hart's lab and a brand new PlantSim model developed from scratch.

One challenge common to each of the models was how to deal with batching an unbatching groups of parts to run them through the printer. One simplification that we could have made was to pre-batch our parts and compute the corresponding characteristics of the batch outside of the model then treat the batch as a single "part" in the process, ignoring the unbatching of the parts at the final post processing step. While this would have yielded reasonable results given the complexity of the model as is, we felt that maintaining the batching and unbatching was important to build into the model structure this capability so more complex batching schemes could be developed down the road. The batching of parts is a critical element to effective utilization of the printing assets but also to the part quality and depends on factors beyond geometry alone (such as orientation, amount of support material, proximity of parts, heat dissipation through the printed material). Maintaining the ability to batch and unbatch parts in the model provides the flexibility to build in more complex batching logic as this knowledge is available.

Both Matlab and PlantSim were effective environments for establishing this model. As with any complex and highly customizable software, the PlantSim software has a steep learning curve. Compounding this issue to the lack of good documentation and examples available online. There was little example code available online to learn the basics of SimTalk and the PlantSim environment is not particularity user friendly for debugging and editing code. PlantSim is relatively simple for single part flow and machine parameters that are not dependent on the part types moving through them. However, as these constraints are relaxed there is inevitably a requirement to begin coding and customization the model. This is both the power and challenge with the PlantSim environment.

The distinct advantage to Matlab is the ability to customize each component of the model. However, this also makes the code error prone and potentially complex as the model grows. The PlantSim environment provides pre-defined attributes to the components of the model. The object oriented approach already defines the shell of attributes need to describe and machine or part. This can be very effective in jump starting the model development if one is familiar with how the software treats each of the different building blocks that make up the model.

PlantSim provides means to generate plots of variables that are tracked in the model. However, the ease of use and flexibility of these plots is limited. In many cases, to obtain the required information, it is often best to export data to a table as the model is running and then use graphics software outside of PlantSim to generate the plots. This is cumbersome and also required some coding expertise to record the data during the simulation. Matlab has stronger functionality for generating and manipulating plots.

Using both modelling approaches – Matlab and PlantSim – provided an effective means to evaluate functionality of model development in both environments. While PlantSim is a difficult software to learn, in part because of a lack of good training documentation, it provides a level of functionality which would be very difficult to replicate in Matlab should the model complexity increase beyond what we attempted here.

IX. Ideas for Future Work

Per Section III where we discuss our original proposal, our ideas for future work revolve around using Shakirov's SLM production model as well as our MATLAB code to not only describe a theoretical line, but to be used as a tool to optimize an additive manufacturing production line.

After exploring both models, questions that we think would be valuable for both the Mechanosythesis Group and to answer in furthering their cost modeling project would be:

1. How robust are PlantSim's worker modeling and are they robust enough to model worker behaviour based on observations from an actual 3D printing line? Because 3D printing lines have less throughput due to the time required for printing, most workers on an additive manufacturing line are cross-trained in most of the processes and are not sitting in one place repeating a process over and over. Can PlantSim accurately model that type of system and what is the benefit to modeling the workers?

2. Can our models be used to create an optimized batching strategy for a 3D printed line. The current batching strategy of max mixed-fill used is a heuristic. Could you input different parts and geometries and determine the ideal batching strategy to minimize the per part costs or reallocate part costs to higher margin accounts?

3. Use PlantSim to optimize the layout of an additive manufacturing line based on a known demand distribution of multiple parts from multipole customers. Also explore how difficult it would be to use our MATLAB code to also optimize the factory's layout.