Metamodeling of Deteriorating Reusable Articles in a Closed Loop Supply Chain

Chapter · August 2021

DOI: 10.1007/978-3-030-85874-2_21

Eoin Glennane
Dublin City University

Metamodeling of Deteriorating Reusable Articles in a Closed Loop Supply Chain

E, Glennane, J. Geraghty
School of Mechanical & Manufacturing Engineering, Dublin City University, Dublin 9, Ireland

Abstract

In this paper I present a closed loop supply chain for a reusable, deteriorating tool. The tool is used in a manufacturing process on an item in a linear supply chain. A model is created for the linear item supply chain and the tools closed loop supply chain to analyse the interactions between them and various input parameters so that output responses of the system can be modelled. Three approaches are taken to model the system, a brute force factorial design, a modified version of a Latin hypercube space filling design, and a fast flexible space filling design. It is found that all three methods can describe responses that require only a few inputs well but cannot accurately predict more complex responses without all of the relevant factors. Space filling designs should be used if more factors are needed as they minimise the total amount of simulations needed to produce an accurate model.

Keywords: Reusable Articles, Deteriorating Article, Closed Loop Supply Chain, ExtendSim, Space Filling Design, Metamodeling.

Introduction

Improving efficiency of supply chains, implementing reverse supply chains and remanufacturing are all topics of research that are becoming more important as the world's resources become scarcer. The desire for companies to embrace this movement has been driven by environmental, social, and financial motivations as more research proves the efficacy of implementing such systems[1].

In this paper, I present a supply chain that produces two items, one high and one low quality, using the same deteriorating tool. The company running the supply chain wants to know how well it is running. To do this a simulated version of the item processing supply chain as well as the tool supply chain is created. The company currently orders new tools to replenish the supply of tools for each line as they are needed but random ordering times can be costly, and less dependable when trying to meet production targets. Instead, an ordering policy for purchasing new tools according to a schedule should be created that aims to meet production targets at a high percentage of the time. To create an ordering policy that will meet its targets and can be adjusted according to multiple input factors is the ultimate goal for the company.

ExtendSim, a program for modelling discrete event, continuous, agent based, discrete rate and mixed mode processes, was used to create and run the simulations. All simulations were conducted on a desktop computer with 32gb of RAM, an ASUS Strix GTX 970 Graphics Card, and an Intel Core i5-9600K CPU @ 3.7GHz. Design-Expert, a statistical software package was used to design the experimental scenarios. Microsoft Excel, JMP (another statistical software analysis package) and Design Expert were used to analyse the results of the simulations. In the final iteration of the model, each simulation took 5-7 seconds to complete and over eight thousand simulations were conducted for a total computational time of roughly 13.3 hours. On a higher specification CPU this time may be reduced.

Objectives

The main objective of the paper is to analyse and compare multiple metamodeling methods applied to a closed loop supply chain (CLSC) that included a reusable article. A sufficiently accurate metamodel of a CLSC could aid in the decision making process and allow a company to predict the outcome of choosing certain production parameters or the effect of setting certain ordering policies without having to simulate each individual scenario which could number in the 1000's or 10'000s and could take hours or days to run for more complex supply chains.

The objective of the metamodeling methods is to minimise the number of simulations needed to produce an accurate metamodel. As a system becomes more complex, the computational time needed to complete a single run increases exponentially hence the need to minimise the total number of simulations. Within each metamodel, the goal is to minimise and maximise certain responses such as the time an item spent queueing or the time between new tool orders, respectively.

Literature Review

Closed Loop Supply Chains are a key component of this study. While the items being processed by the model are not in a closed loop, the tools used to process the items can be classed as a reusable product[2] and also a deteriorating product[3] inside a closed loop supply chain. S.Singh et al.[4] explored a mathematical approach to a very similar problem in which remanufacturing of a pair of deteriorating items of two different qualities was integrated into a closed loop supply chain. This paper will be taking a simulation based approach.

Metamodeling has become an important tool for operational efficiency in many different types of industries, from design of vegetative filter strips[5] to satellite visibility prediction[6] and more. With the goal of simulating how a complex system reacts to inputs accurately and quickly, metamodeling, if conducted correctly, can output a simple model of a system, and remove the need to conduct further simulations. The Latin

Hypercube design has been shown to work with this type of model, one that includes random demand while minimising system outages[7]. Factorial designs are useful and quick at examining models with multiple independent variables[8] however the total amount of simulations can increase significantly as a model becomes more complex.

Model Development

With the goal of examining how a reusable article interacts with a supply chain, model development began in ExtendSim with the linear, forward supply chains for the items.

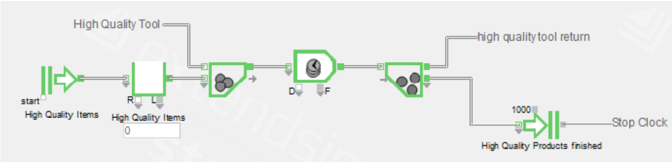

Figure 1 – Item Supply Chain First Iteration

The model as a whole changed many times throughout the course of model development as more about the interactions between certain aspects of the model were understood. The mechanisms by which the tool returned to the supply chain for use changed multiple times over the course of the model development but always kept to a similar pattern.

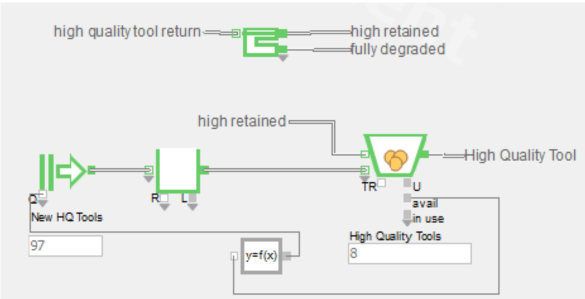

Figure 2 - High Quality Tool Queuing First Iteration

The tool would be kept in a queue until an item entered the queue. The tool and item would leave the queue together, the item would be processed while the tool waited in another queue for the item to finish processing. Once the item was finished being processed it was again, released from the second queue with the tool, at which point the item would exit the supply chain, having been processed, while the tool was redirected to be artificially deteriorated. Depending on the numerical value for the tool's quality post degradation, the tool would then be sent back to its original queue to wait for another item, be sent to an identical queue to be used in a lower quality production line or discarded entirely.

The problems with this original model stem from the lack of flexibility in the blocks used to simulate the queue, so other methods of tool simulation had to be explored.

One of the factors that was initially chosen to be explored in more detail was the use of an operator i.e. a simulated worker. However, in the preliminary testing phase, based on the number of operators assigned at the beginning, problems were encountered where operators would be duplicated based on certain actions within ExtendSim at random. It was decided after a few weeks of testing to keep the number of operators equal to the number of machines in the system for all further simulations. In this scenario, the duplications did not occur but the effect of having more or less operators than machines could not be explored.

The first factor implemented into the model was the Initial Number of Tools supplied by the system at the beginning of each run. This was accomplished using a set block to send the set number of items into the system at initialisation according to the run criteria.

The model went through three major iterations, being finalised once all of the factors chosen could be input from a database into the system correctly and the responses of the system were calculated and input into a database correctly. All versions of the models are available in the detailed set of appendices alongside this document.

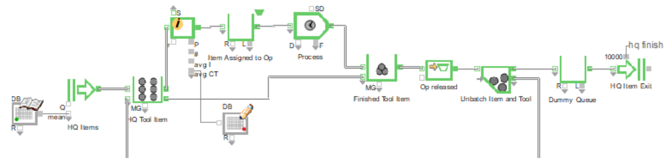

Figure 3 - Finalised Item Supply Chain

The final iteration of the model builds utilised Queue Matching and Unbatching blocks in ExtendSim to achieve the necessary flexibility for programming the factors and recording the required responses in the model.

Item creation was a factor that seemed to have the most effect on the responses in the system and was one of the hardest areas to balance so that all factors and responses could be measured adequately. Process Time for the Machine was set to a lognormal distribution with a mean of 1 and a standard deviation of 0.1 while the distribution for the "Create" Block was an exponential distribution that was varied over the simulations between a mean of 1.1 and 1.25. The Lognormal distribution has been shown to model activity time[9] in real scenarios while the Exponential distribution has been shown to model interarrival times in processes such as the one in this model[10] in industry settings.

One of these responses is measured by the "Information Block" directly after the "Queue Matching" block labelled "HQ Tool Item" in Figure 3. The "avg CT" output of the information block measures the average time that an item spends waiting in the queue before it is assigned a tool and can be processed.

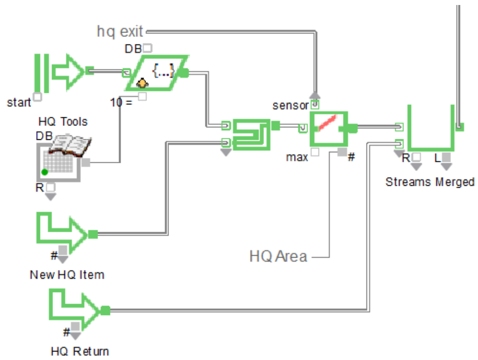

Figure 4 - Tool Creation and Return

The number of tools currently in the system was tracked using a “Sensor" block. With the sensor block located just after tool creation, it is able to track how many items have entered the supply chain, while the “hq exit” variable tracks the number of tools that have left this specific supply chain. When the number tracked by the sensor incremented below the limit set by the user or by the scenario criteria, a signal would be sent to purchase new tools for the system.

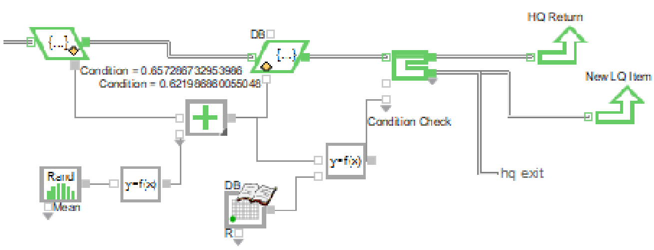

Figure 5 - Tool Degradation

When each tool is detached from a processed item, it goes through the degradation and return process. The “Get” and “Set” blocks, accompanied by math and equation blocks, simulate the degradation of the tool by incrementing the condition variable of each individual tool that passes through by a certain amount. Once a tool is degraded, its new condition variable is checked, and the tool is either returned to its original pool of tools or sent to the low quality production line. For example, if a slightly used tool entered the “Get” block with a value of 0.62 and was degraded by 0.05 to a value of 0.57, and the cut-off point for high quality tools was 0.6, then the tool would be sent on the downward path to be used in the lower quality production chain. This is the action that would cause the sensor block to increment its score downwards by one. One of the criteria that was chosen as an important variable was the cut-off point at which a tool was demoted to the other supply line or discarded altogether.

Another factor that could have been implemented into the model but was not, is the amount by which a tool is degraded every time it is used. For both supply chains, each time a tool was sent down the degradation line, the unique condition value attached to the tool was degraded according to a lognormal distribution with a mean of 0.1 and a standard deviation of 0.1. With an initial quality level of 1, a HQ cut-off value of 0.6 and a LQ cut-off value of 0.2, the average tool would be used four times on each production line before being scrapped.

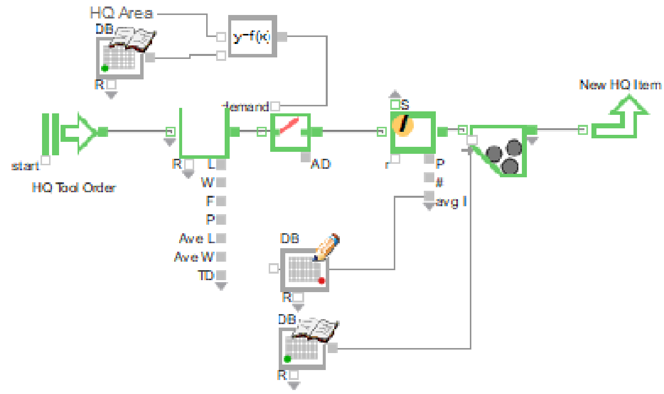

Figure 6 - New Tool Creation

Creating new tools for the system is the area that took the greatest number of iterations to complete fully before it behaved in a way that could be analysed correctly and easily. The sensor block is in constant communication with this area of the system. Whenever its reading for “number of items currently in the area” goes below a threshold as set by the user/scenario, a signal is sent to send new tools into the supply chain to be used. Using another sensor block, an information block and an “Unbatching” block, the system, when it senses this signal it opens the gate for a single tool to be allowed through. These single tools passed through the information block which measures the average amount of time between each of the orders (The second response for the system), then passes through the “Unbatching” block, splitting the tool into multiple tools according to a variable set by the user/scenario. This factor of the number of tools the individual tool is split into is referred to as the Batch Size.

The number of factors added to the model was initially eight, however this was further reduced to six to reduce the total amount of simulations for the “brute force” method of metamodeling in Design Expert and the two factors that were cut did not seem to have a major effect on any of the responses after some preliminary testing. The original eight factors being Initial number of High Quality Tools, Initial number of Low Quality Tools, Low Inventory Limit, High Quality Tool Cut-off Point, Low Quality Tool Cut-off Point, Batch Size, Mean High Quality Items, and Mean Low Quality Items. The Initial number of tools for both high and low quality tools had little to no effect on any response, so it was amalgamated into a single factor labelled “Initial Tools”. The same process was conducted with the Mean High and Low Quality Items as preliminary testing revealed that while the mean high quality item variable had a large effect on all of the model, the low quality item variable had little to no effect on any of the model. Figure 7 is an example of some of the runs conducted and their respective factors in a database that the model refers to for each run.

	Init Tools[1]	Low Inv Limit[2]	HQ Tool Cutoff[3]	LQ Tool Cutoff[4]	Batch Size[5]	Mean Items[6]
85	8.00	1.00	0.40	0.10	4.00	1.1632
86	4.00	2.00	0.60	0.10	5.00	1.1609
87	12.00	1.00	0.50	0.00	5.00	1.1564
88	14.00	1.00	0.30	0.20	4.00	1.1587
89	14.00	2.00	0.30	0.10	6.00	1.1519
90	8.00	2.00	0.30	0.00	3.00	1.1542
91	6.00	4.00	0.60	0.20	3.00	1.1496
92	12.00	4.00	0.40	0.20	3.00	1.1472
93	4.00	2.00	0.40	0.00	2.00	1.2221
94	14.00	4.00	0.40	0.10	6.00	1.2196

Figure 7 - Factor Database Example

The limits of each of the factors were decided based on human observation of the model, considering the response of the system.

Table 1 - Factor Limits

Factor Limits	Low	High
Init Tools	4	15
Low Inventory Limit	0	4
HQ Tool Cut-off	0.3	0.6
LQ Tool Cut-off	0	0.2
Batch Size	1	6
Item Mean	1.1	1.25

Choosing responses for the system to conduct the sensitivity analysis with was very limited as no cost variable was associated with any stage of the cycle. Four responses were chosen to be analysed in the final iteration of the model, two for each product line, High and Low Quality production lines. The first, “Time Between Orders” measured the average time between orders for a new batch of tools entering a product line. This response did not have a set goal such as minimisation or maximisation as, in the context of an industry setting, as long as you can predict what the average time between orders will be, you can set up an ordering policy in advance that will meet the needs of the system a high percentage of the time.

The second response measured was “Item Queue Time” which output the average time that an item spent waiting for a tool before being able to be processed. The goal for this response is to minimise, as the shorter time the item spends waiting for a tool, the shorter total lead time the item will have, meaning more items can be processed in the same amount of time. While this response also helps a company estimate how long an item might spend waiting, it can also act as a performance indicator for a particular ordering policy or supply chain setup. Longer wait times for items may indicate that the ordering policy for tools has to be adjusted to meet demand.

Metamodeling

With the limits for the factors chosen and the responses ready to calculate, the metamodeling could begin. In total, two iterations of metamodeling occurred. The first included both the high and low quality initial tool limits while the second iteration only included a set “Initial Tools” limit which was used by both the high and low quality supply lines. The first iteration also had both high and low quality item means but that was also changed to just one item mean for the second iteration as discussed previously.

The reduction of the number of factors included in the model reduced the number of simulations required for the Design Expert metamodel significantly. As the design expert model is a factorial one, the number of simulations required by the model is equal to two to the power of the number of factors. With eight factors, 256 simulations have to be run to create that model. However, to reduce any outlier and randomness in the model, it was decided that ten repetitions of each scenario had to be run and have the results averaged. In total 2560 simulations had to be run for this model to be created. The reduction of the number of factors from eight to six reduced the total number of simulations needed from 2560 to just 640, a reduction of 75%.

This adjustment of the number of factors also had an effect on both of the JMP space filling metamodel designs. Both space filling designs had the number of simulations required lowered from 800 to 600 (These numbers include 10 repetitions of each scenario). A reduction of 25%.

The first metamodel created in JMP was a Latin hypercube based design. This type of design has been shown to produce accurate metamodels and assist in the decision making process when multiple variables and responses are taken into account[11]. One problem that was encountered in this path was that the Latin Hypercube generator in JMP only allowed continuous values for each of its factors. While testing this data in ExtendSim, many problems occurred with duplication of items and tools when a factor such as batch size had a non-integer value attached, so changes to the Latin Hypercube Model had to be made. Each of the factors that required only integer values had its generated results rounded to the nearest integer, simulating that the data was actually categorical. Table 2 depicts the data types required for each factor to avoid errors in ExtendSim.

Table 2 - Factor Data Types Required in ExtendSim

Factor	Data Type
Initial Tools	Categorical
Low Inventory Limit	Categorical
HQ Tool Cut-off	Continuous
LQ Tool Cut-off	Continuous
Batch Size	Categorical
Item Mean	Continuous

While this may have compromised the integrity of the metamodeling method, the other option was to not use the method altogether. This modified approach did not contain any of the duplication errors and the results produced were error free in initial testing. I refer to this method from here on as the Rounded Latin Hypercube (RLHC) method.

The Fast Flexible Filling Design (FFFD) in JMP allowed for categorical data alongside continuous data so no cleaning of the input factors for the simulations was required. This was chosen alongside RLHC for comparison.

The first iteration of each of the metamodeling methods used an item creation mean range of 0.8-1.4 with an exponential distribution pattern however it was observed in the results that as the mean tended towards 1, the wait time for the tools increased almost exponentially and all interactions of the other factors were completely overshadowed by the mean. In the second iteration, this range was refined to 1.1-1.25 so that the other factors could be analysed alongside the mean item time.

Simulation Setup

Databases were created in ExtendSim so that any number or combination of runs could be setup to simulate back to back. Once a metamodel was created in JMP or Design Expert, it was exported to Excel for adjustment if necessary. The run specifications were then duplicated nine times within Excel and then pasted into the database in ExtendSim. The amount of runs that ExtendSim conducted was always equal to the number of rows in the factors database so once the run button was pressed, ExtendSim ran all the simulations required, 640 for Design Expert, and 600 for both JMP models in the final iteration. The data recorded in each run was written into another database for the responses of the system at the end of each run. At the end of a full set of runs, the output would be a second database with the same number of rows as the number of runs/simulations. This data was exported to excel where each of the ten duplicated runs were average. The averaged results could then be exported back into Design Expert and JMP as needed and analysing of the results were conducted.

Results

The aim of using the metamodels is to create an accurate model of the system while minimising the amount of computation needed to make it accurate. As each of the three methods needed roughly the same amount of simulation time (600-640 runs each) to complete their models in the final iteration, we can say they all perform the same in this aspect. The way in which they must be compared is then by the accuracy of their models for each of the responses.

Comparison of the R-squared values for each of the models for each of the responses was conducted and tabulated in Table 3. An R-squared value above 0.90 is deemed to be accurate and can be classified as successfully emulating the simulation, provided it is not disqualified in the validation stage.

Table 3 - R-Squared Results

R-Squared Values	FFFD Predicted	RLHC Predicted	DE Predicted
HQ Time Between Orders	0.9214	0.9342	0.9948
LQ Time Between Orders	0.4526	0.5797	0.9624
HQ Item Queue Time	0.9214	0.9342	0.9081
LQ Item Queue Time	0.4526	0.5797	0.8865

All three methods seem to predict the high quality time between orders and the high quality item queue time very well but only the design expert method has a high R-squared value for the low quality production line.

Table 4 - Metamodel Validation

	Simulated Results	FFFD Predicted	RLHC Predicted	DE Predicted
HQ Time Between Orders	28.84	29.09	28.86	29.70
LQ Time Between Orders	201.70	116.99	294.97	43.50
HQ Item Queue Time	0.95	1.18	1.18	1.71
LQ Item Queue Time	0.41	0.52	0.48	0.43

Validation of the models was conducted by taking a random model scenario and running it ten times to get an average result and comparing each of the model's predictions to the averaged result of the simulation. The ten simulations were run with the following factors: Initial Tools = 10, Low inventory Limit = 2, HQ Tool Cut-off = 0.45, LQ Tool Cut-off = 0, Batch Size = 4, Item Mean = 1.175. In Design Expert, the confirmation tab allows the user to input numbers for each factor and will output the expected responses. In JMP, a profiler was constructed using a partial least squared regression method for both JMP models that allowed the user to input values for the factors and receive estimated responses.

Discussion

Modelling the HQ time between orders seemed to be the one area that all three models successfully completed, all yielding R-squared values greater than 0.9 as well as all having accurate (<5% error) estimations in comparison to the averaged simulation results. The other three responses were not as close in estimation except for the Design Expert model in predicting the low quality queue time response. This estimation only yielding an error of 3.6% in comparison to the simulations while the RLHC and FFFD responded with errors of 15.4% and 25.6% respectively.

Estimating the low quality time between orders was the worst predicted response for all models with the best model (FFFD) yielding a value 42% lower than the simulated value while the RLHC and Design Expert Model estimated +46% and -78% of the simulated value. Even though the Design Expert model had an R-squared value that indicates that is has produced an accurate model, the predicted value is very far off the simulated result. The same phenomenon occurred with the design expert model for predicting the high quality item queue time with an R-squared value of 0.9081 yet the estimate is 79.7% greater than the simulated value. The RLHC and FFFD both yielded values ~24% greater than the measured value for high quality item queue time.

The models producing good results for “HQ time between orders” in comparison to the other three responses may be due to the number of variables that influence the results. For example, the high quality item line only takes item inputs from one area and new tool inputs from one area, whereas the low quality line takes item inputs from one area but takes new tool inputs from two areas, one being controlled by an ordering policy while the other is semi-random inputs of tools from the high quality line. Figure 8 illustrates this type of occurrence in the lower quality line as a group of tools all get demoted to the lower line in a small time frame, resulting in a huge number of tools and a large time frame for the tools to diminish again.

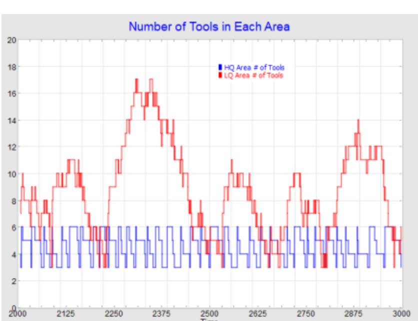

Figure 8 - Tool Tracking Mid-Simulation

This type of influx into the low quality line is something that none of the models consider. I believe, that if another factor were added to the models that represented this behaviour in the system, the results could be more accurate.

The RLHC model for the low quality responses was poor. With an R-squared value of 0.5797 for both LQ responses, some important factor is clearly missing. While some factors such as initial tools, low inventory limit and batch size had little to no effect on each response, factors such as mean items and HQ/LQ tool cut-off had a significant impact.

Model Coefficients for Original Data

Coefficient	LQ Time Between Orders	LQ Item Queue Time
Intercept	73.4025	13.4347
Initial Tools	-6.3474	-0.0008
Low Inventory Limit	24.8726	-0.1773
HQ Tool Cutoff	1533.3533	-2.7542
LQ Tool Cutoff	-762.4769	1.8449
Batch Size	59.2238	-0.1502
Mean Items	-588.6050	-9.1537

Figure 9 - LQ Response Analysis for RLHC

I think there are a few areas of the model that could be looked at to address this issue, the first being the rate at which tools leave the high quality area and enter the low quality area. A linear rate of addition of tools to the LQ tool supply could be an example of a factor that could be incorporated into the model specifically for the low quality responses.

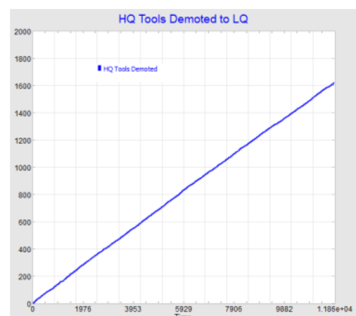

Figure 10 - Linear rate of demotion of HQ Tools

A similar approach could be applied to the FFFD metamodel as well, as it shares similar R-squared values and similar results.

Randomness in the ExtendSim simulation may also be a contributing factor to the lack of accuracy of the metamodels. While repetition of simulations was conducted for each model, outliers still seemed to have an effect on the responses chosen for the system. Item generation, Item process time, and the rate of tool degradation were all influenced by random numbers while also being three areas of the model with a huge influence on results. While only the mean item generation was included as a factor in each of the models, adding these three areas of randomness as factors to the models could produce more accurate results in the metamodeling process.

Equations to calculate each of the responses for each of the metamodeling methods based on the input factors provided were created. In this paper, I will only be listing the three equations for each "HQ Time Between Orders" as they were the most accurate out of the twelve results. The remaining equations can be found in the appendices for this paper.

Table 5 - Design Expert HQ Time Between Orders Equation

Multiplier	Factor
-0.082	Intercept +
0.089	*HQ Tool Cut-off +
4.54	*Batch Size +
0.03	*Mean Items +
-10.17	*HQ Tool Cut-off *Batch Size +
6.35	*Batch Size *Mean Items

Table 6 - RLHC HQ Time Between Orders Equation

Multiplier	Factor
-8.13	Intercept +
-0.036	*Initial Tools +
-0.045	*Low Inventory Limit +
-38.79	*HQ Tool Cut-off +
7.16	*Batch Size +
22.34	*Mean Items

Table 7 - FFFD HQ Time Between Orders Equation

Multiplier	Factor
-13.69	Intercept
24.86	*Mean Items +
7.41	*Batch Size +
-37.46	*HQ Tool Cut-off +
0.051	*Initial Tools +
0.14	*Low Inventory Limit

Conclusions

All three models, Design Expert, RLHC, and FFFD, accurately predict the response with the least amount of input: High Quality Time between Orders. However, the more complex responses of the system are not accurately captured in any of the three models. One of the reasons for this may be the lack of relevant factors analysed by the simulation. Another reason could be that there is too much unaccounted randomness in the system for the models to accurately predict the more complex performance indicators. The use of space filling designs such as RLHC and FFFD perform well with high numbers of factors and should be explored further. The Design Expert model performs well but the number of simulations required increases exponentially with the number of factors present.

Further Work

Only two types of space filling designs for metamodel creation were explored in this paper, a modified version of the Latin Hypercube and JMPs Fast Flexible Filing Design. Both methods are one-shot, non-sequential designs so sequential methods[12] may be able to create accurate models with even lower numbers of simulations required.

References

[1] Mastos, T. D. et al., 2019, "Introducing an Application of an Industry 4.0 Solution for Circular Supply Chainmanagement," Sci. Total Environ., p. 135907.

[2] Carrasco-Gallego, R., Ponce-Cueto, E., and Dekker, R., 2012, "Closed-Loop Supply Chains of Reusable Articles: A Typology Grounded on Case Studies," Int. J. Prod. Res., **50**(19), pp. 5582–5596.

[3] Moubed, M., Boroumandzad, Y., and Nadizadeh, A., 2021, "A Dynamic Model for Deteriorating Products in a Closed-Loop Supply Chain," Simul. Model. Pract. Theory, **108**, p. 102269.

[4] Singh, S. R., and Saxena, N., 2013, "A Closed Loop Supply Chain System with Flexible Manufacturing and Reverse Logistics Operation under Shortages for Deteriorating Items," Procedia Technol., **10**, pp. 330–339.

[5] Lauvernet, C., and Helbert, C., 2020, "Metamodeling Methods That Incorporate Qualitative Variables for Improved Design of Vegetative Filter Strips," Reliab. Eng. Syst. Saf., **204**(February 2019), p. 107083.

[6] Wang, X., Han, C., Yang, P., and Sun, X., 2019, "Onboard Satellite Visibility Prediction Using Metamodeling Based Framework," Aerosp. Sci. Technol., **94**, p. 105377.

[7] Chen, Q., Zuo, L., Wu, C., Bu, Y., Huang, Y., Chen, F., and Chen, J., 2019, "Supply Adequacy Assessment of the Gas Pipeline System Based on the Latin Hypercube Sampling Method under Random Demand," J. Nat. Gas Sci. Eng., **71**(July), p. 102965.

[8] Haerling (Adamson), K., and Prion, S., 2020, "Two-by-Two Factorial Design," Clin. Simul. Nurs., **49**, pp. 90–91.

[9] Trietsch, D., Mazmanyan, L., Gevorgyan, L., and Baker, K. R., 2012, "Modeling Activity Times by the Parkinson Distribution with a Lognormal Core: Theory and Validation," Eur. J. Oper. Res., **216**(2), pp. 386–396.

[10] Helbing, D., Treiber, M., and Kesting, A., 2006, "Understanding Interarrival and Interdeparture Time Statistics from Interactions in Queuing Systems," Phys. A Stat. Mech. its Appl., **363**(1), pp. 62–72.

[11] Ben Ali, M., D'Amours, S., Gaudreault, J., and Carle, M. A., 2018, "Configuration and Evaluation of an Integrated Demand Management Process Using a Space-Filling Design and Kriging Metamodeling," Oper. Res. Perspect., **5**, pp. 45–58.

[12] Crombecq, K., Laermans, E., and Dhaene, T., 2011, "Efficient Space-Filling and Non-Collapsing Sequential Design Strategies for Simulation-Based Modeling," Eur. J. Oper. Res., **214**(3), pp. 683–696.

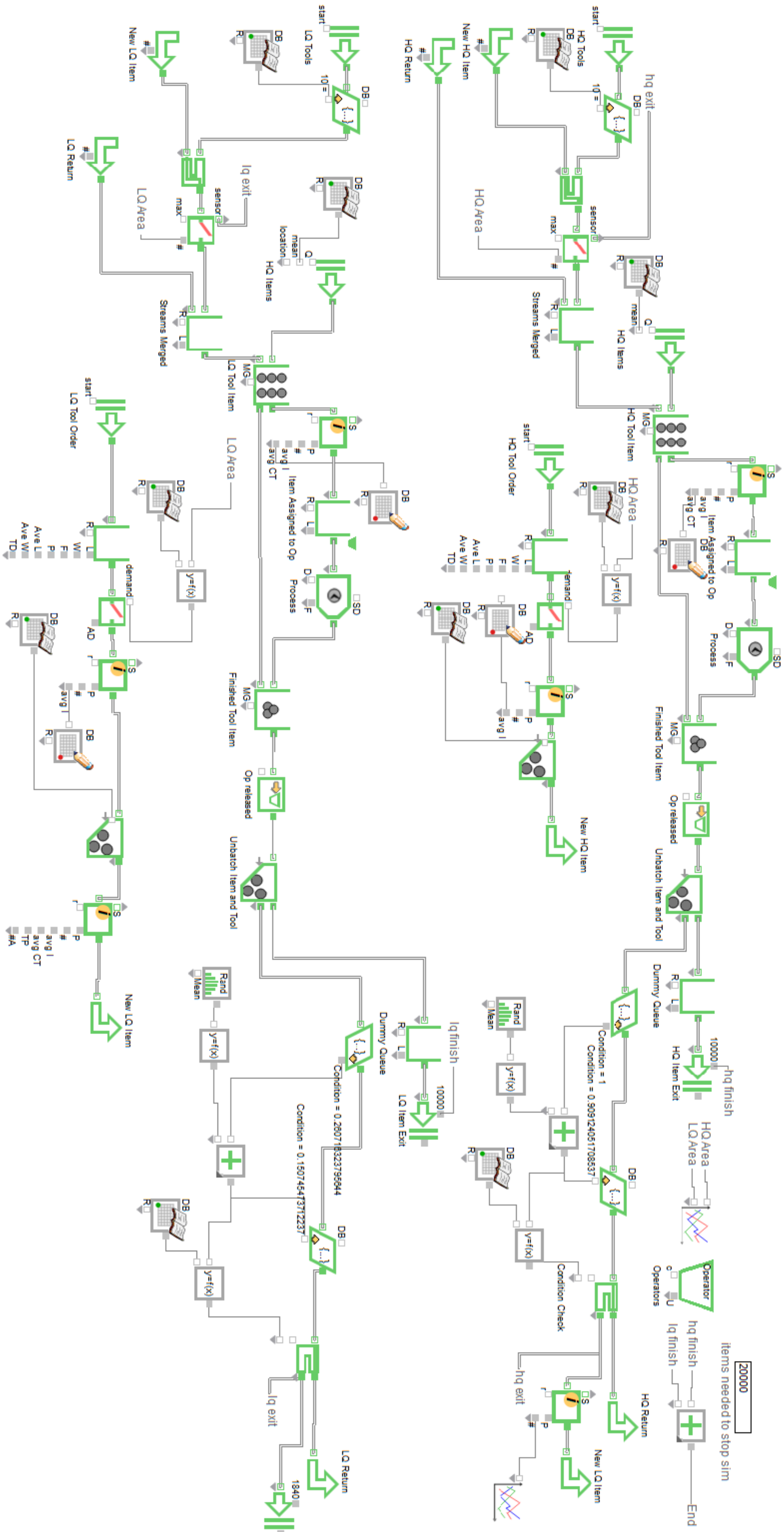

Figure 11 - Final ExtendSim Model