

**SYSC 4005**

Discrete Simulation/Modeling

**Final Report**

**Date:** April 11<sup>th</sup>, 2021

**Carleton University**

**Ottawa, Ontario**

## Table of Contents

<b>1.0</b>	<b>PROBLEM FORMULATION</b>	3
<b>2.0</b>	<b>OBJECTIVE</b>	3
<b>3.0</b>	<b>MODEL CONCEPTUALIZATION</b>	4
3.1	Entities	4
3.2	Essential Features	5
3.3	Assumptions	6
<b>4.0</b>	<b>MODEL TRANSLATION</b>	6
<b>5.0</b>	<b>DATA COLLECTION AND INPUT MODELING</b>	7
5.1	Inspector 1	8
5.2	Inspector 2	10
5.3	Inspector 3	12
5.4	Workstation 1	14
5.5	Workstation 2	17
5.6	Workstation 3	19
5.7	Input Implementation	21
<b>6.0</b>	<b>MODEL VERIFICATION</b>	21
6.1	Events	22
6.2	Inspectors	22
6.3	Buffers	23
6.3.1	Buffer Bounds	23
6.4	Workstations	23
6.5	Products to Components Ratio	23
<b>7.0</b>	<b>MODEL VALIDATION</b>	24
7.1	Results Comparison	24
7.2	Sensitivity Analysis	24
7.3	Historical Data as a Validation Alternative	25
<b>8.0</b>	<b>PRODUCTION RUNS AND ANALYSIS</b>	26
<b>9.0</b>	<b>ALTERNATIVE OPERATING POLICY</b>	28
9.1	Alternative Component 2 Routing	29
9.2	Alternative Component 1 Priority	31
<b>10.0</b>	<b>CONCLUSION</b>	32

## **1.0 PROBLEM FORMULATION**

At this manufacturing facility three types of products are created (P1, P2, P3), each requiring at least one or more component (C1, C2, C3) for its creation. Each product has a specified workstation associated with it, and only that workstation is responsible for that item. Before assembling the product, each workstation must wait for the required components, utilizing a buffer zone that has a maximum capacity of two for each assembly piece. Before reaching the buffer, two inspectors check the components and place them in the specified workstation only if the buffer zone is not full. Inspector 1 chooses the buffer zone with the least number of components waiting in the queue. In the event of a tie, Workstation 1, 2, then 3 is favored. Looking at this problem, a simulation appears to be the best method of determining the most ideal workflow as the different stations, priority, and components will create a significant amount of potential system states, where the ideal workflow will be difficult to determine mathematically. From the initial appearance it seems very likely that Workstation 1 will have higher rate of production than Workstation 2 or 3 as it only requires one component, disregarded each Workstation's processing time. In event of a tie, Workstation 1 will possibly be lower in the priority list, as the increased priority will increase the production rate in addition to the already increased production from the one component requirement.

## **2.0 OBJECTIVE**

The objective of simulating this problem is to identify the ideal behaviour of Inspector 1, and to determine the ideal priority of each Workstation in distributing Component 1 in a manner that benefits all product production. To determine the ideal workflow, Inspector 1's criteria to

distribute Component 1 must be changed, as well as the priority of each Workstation when a tie occurs.

### 3.0 MODEL CONCEPTUALIZATION

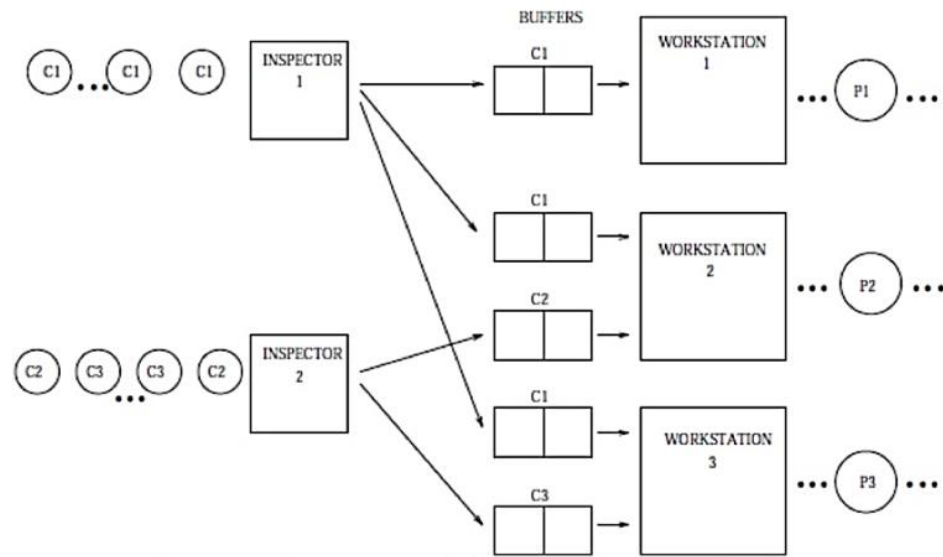


Fig.1: Schematic illustration of the manufacturing facility.

#### 3.1 Entities

**Products:** The resultant output of each workstation, in the defined problem. Three potential products exist (P1, P2, P3).

**Components:** The building blocks of each product and the recipe of components for each individual product (P1, P2, P3) differs and production cannot occur without the required components (C1, C2, C3). The supply of components is unlimited and will not slow down the production process.

**Inspectors:** Each component is inspected by two Inspectors, each responsible for their own set of components (Inspector 1 = C1, Inspector 2 = C2, C3). The inspector is responsible for placing the components into the buffer zones adjacent to the workstations.

**Workstations:** This is where the components are assembled into products. Each product (P1, P2, P3) has a workstation associated with it (W1, W2, W3) where the respective product is assembled.

**Buffer Zone:** The component queue for the workstations to produce products. Each workstation can only hold a maximum of two components each, a total of five buffer zones exists for each product (P1, P2, P3).

### **3.2 Essential Features**

The essential features of the system are the ones that limit the rate of production, which we will consider to be the inspectors, workstation, and buffer zones. The time each inspector takes to inspect each component is crucial, the differing times can change how we prioritize components for each workstation. However, while we consider the workstations as essential, they will only compose products at the rate defined in the data and can only be affected by the components supplied. No way exists to increase the rate the workstations process components other than supplying consistent assembly material. If the components are readily available in the buffer, the products will always be produced at the rate defined for each workstation. However, the processing times for each workstation are essential in the priority of component supply they

will receive. Since the components are unlimited, we cannot consider it an essential feature or a bottleneck for the system, as it is readily available.

### **3.3 Assumptions**

The first assumption of the system we make in order to simulate the system is the behaviour of Inspector 2. Since the Inspector 2 is responsible for two components, we expect the Inspector to select between Component 2 or 3 at random or some defined interval stated in the data set. Secondly, if the Inspector is blocked from placing components as the buffer zones are full, we reasonably expect Inspector 2 to move onto another component that is needed or if all buffer zones are filled hold onto the component without resuming a new inspection (this will apply to both inspectors).

## **4.0 MODEL TRANSLATION**

We have chosen to use the scripting language MATLAB to develop our simulation model. MATLAB has a discrete-event simulation framework (Simulink) that will be used to facilitate the translation of the model into a simulation. It is a common programming language that is often used to machine learning, simulation and designs that require multiple iterations that simply cannot be computed by hand. It is also a language we are familiar with and one we employ in our 4<sup>th</sup> year projects. Simulink can compute the complex math and simulation required for the modules.

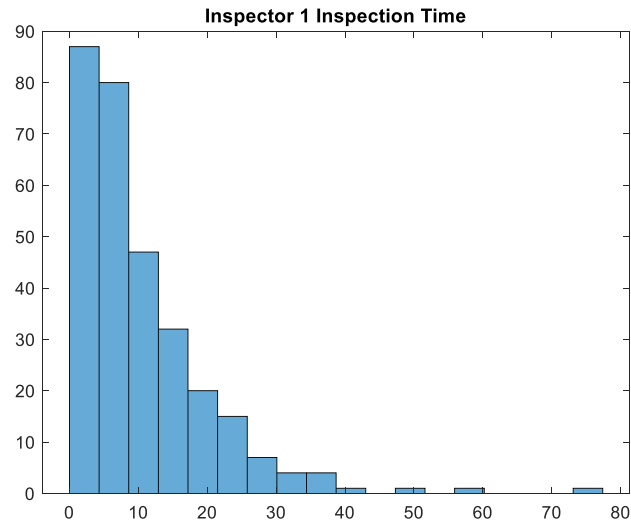
The distributions provided for the inspection times, component selection, workstation processing times and will be randomly sampled to create a distinct simulation of random events to test our potential workflow implementation. A script will be used to determine the characteristics of those random events to process them in way that will output statistical information about the quality and efficiency of our choices. Each potential event will be processed sequentially and as they occur. Events will be processed for each component with multiple checks for buffer limitations and workstation production. Events will be generated to indicate that a workstation has completed its product in order to advance to the next randomly generated event. Inspector idle times for when buffer queues are full will also be considered, this event can/will only be changed once a new product has been completed (new event) as that is the only way for buffer queues to clear up. Each event will only occur at the expected start time and our implementation will not attempt to create future events at any time, considering that the buffer queues will stay in the expected range of zero to two as only causal events will be considered. A script or addition will be created to validate the results from the simulation to determine that they are acceptable and not outlandish. This step is important as it will allow us to debug and create a simulation that is as accurate as possible. The simulation in the early stages will have many errors and we are not expecting it to be accurate to start. Identifying those errors and inaccuracies is key to testing and translating the model into a simulation.

## **5.0 DATA COLLECTION AND INPUT MODELING**

Historical data of the inspectors' and workstation' service times for each respective inspector and station are used to simulate and assess the performance of the manufacturing facility. The inspection and service times will be evaluated to identify each distribution and will

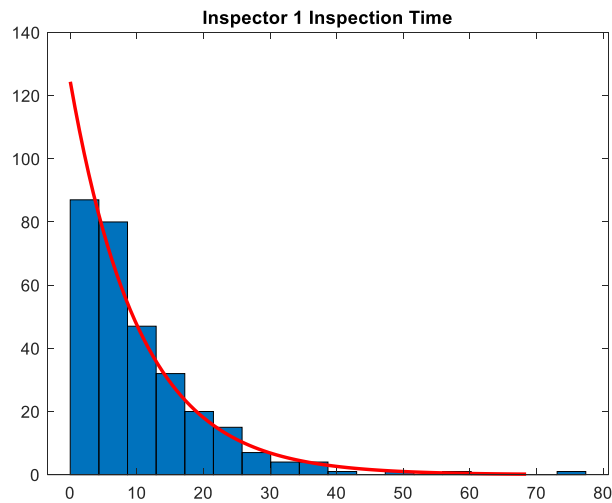
be validated. Each data set has 300 elements and therefore will have 18 bins in each respective histogram.

## 5.1 Inspector 1



**Figure 1: Histogram of Inspector 1's Historical Service Times (18 bins)**

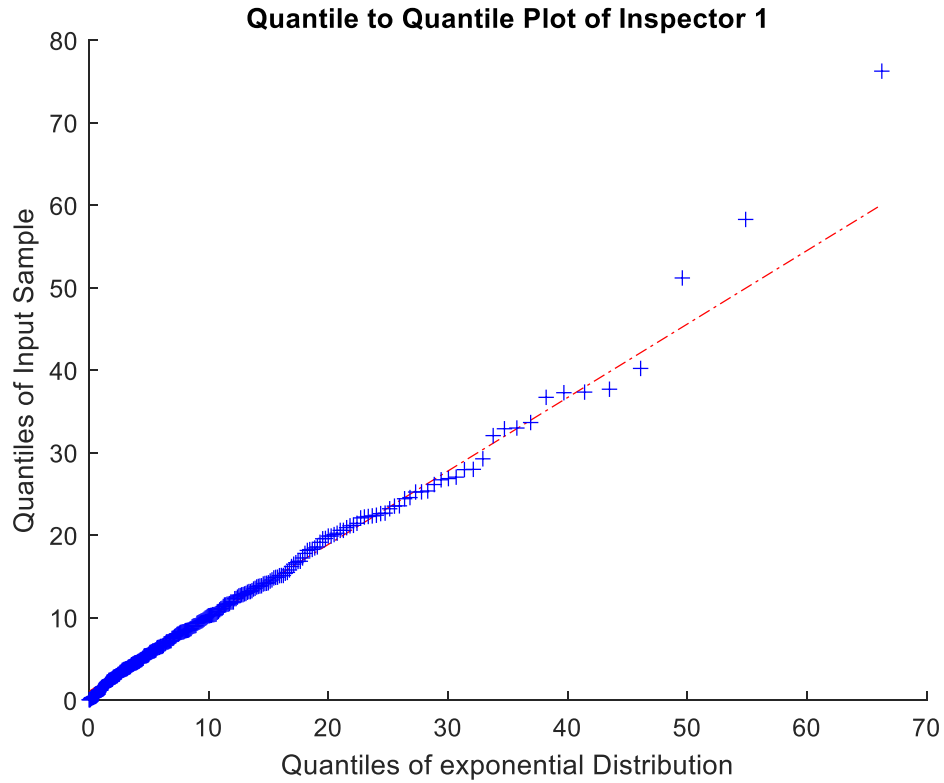
From the initial appearance of the histogram (Figure 1), it is clear the historical data set of Inspector 1 follows an exponential distribution.



**Figure 2: Inspector 1's Histogram with an Exponential Fit**



After overlaying an exponential function over the histogram (Figure 2) and it becomes even more obvious that the data set follows an exponential distribution. However, to be certain a chi-square goodness-of-fit test is performed with a quantile-quantile plot to determine if the data follow an exponential distribution.



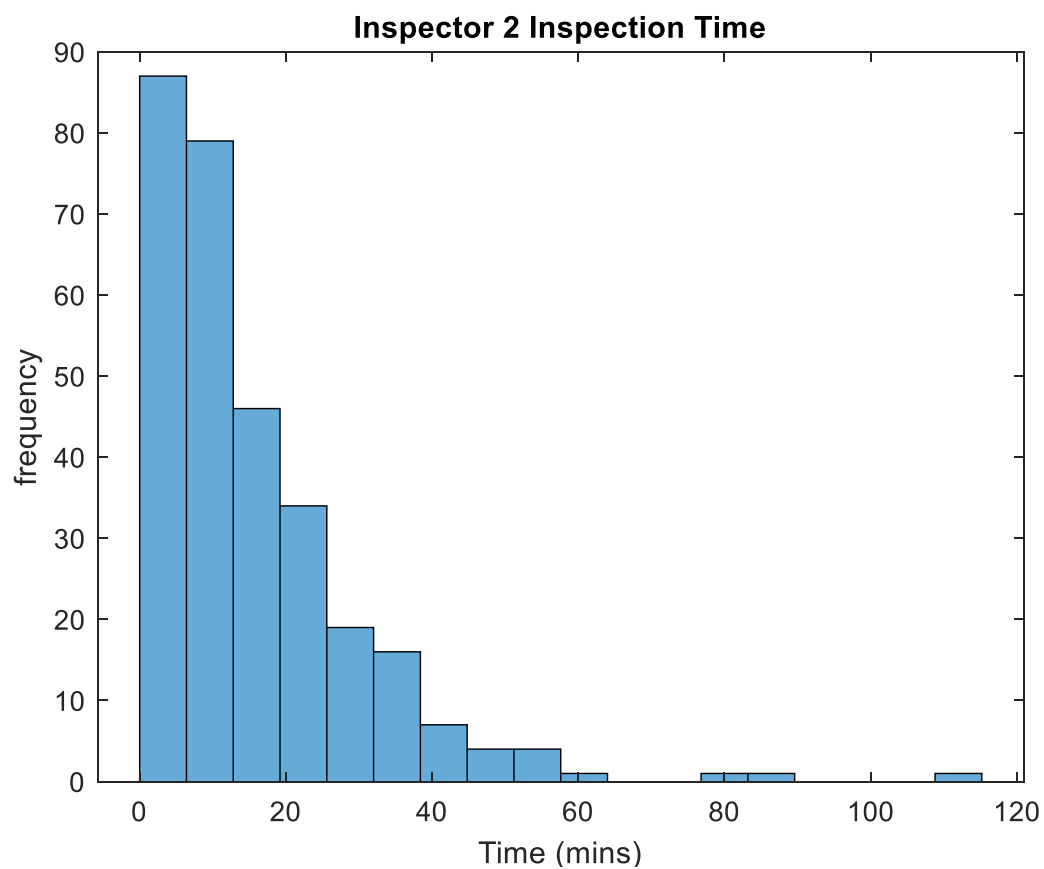
**Figure 3: Q-Q Plot of Inspector 1**

Here can be seen in see (Figure 3, the linearity of the plot confirms that the exponential distribution is a good fit. Some outliers exist near the 55-69<sup>th</sup> quantiles of the distribution, but the overall the fit remains good. The parameter estimator used for this quantile-to-quantile graph is  $\lambda = \frac{1}{x}$  with a quantile function =  $X\ln(1 - x)$ .

The chi-square goodness-of-fit test will determine whether to reject or accept the null hypothesis that the data set follows an exponential distribution with an estimator of  $\lambda =$

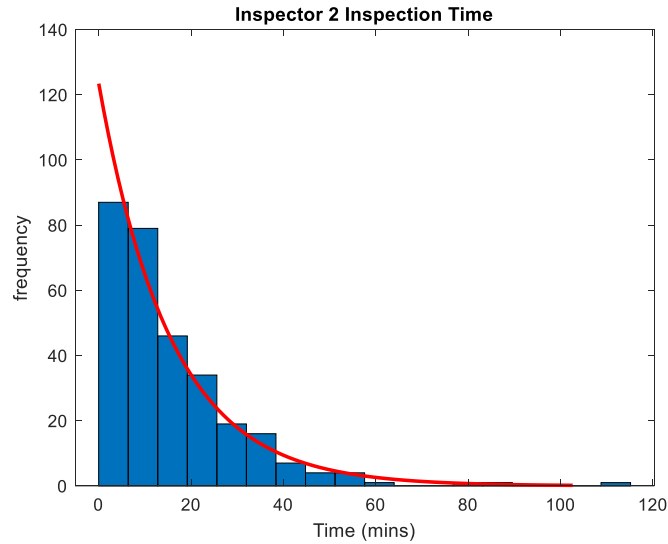
0.09654. With a  $\alpha = 0.05$ , and with 16 degrees of freedom ( $18-1-1 = 16$ ) we obtain the critical value of 7.962. The chi-squared test statistic is 5.2902, therefore we do not reject the null hypothesis and we accept the data set does follow an exponential distribution at a significance of 5%.

## 5.2 Inspector 2



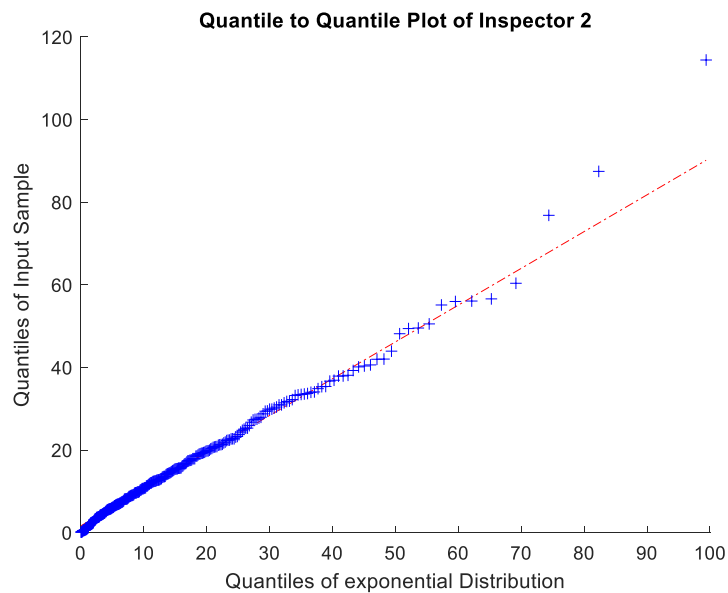
**Figure 4: Histogram of Inspector 2's Historical Service Times (18 bins)**

From the initial appearance of the histogram in Figure 4 it is clear the historical data set of Inspector 2 follows an exponential distribution.



**Figure 5: Inspector 2's Histogram with an Exponential Fit**

After overlaying an exponential function over the histogram, seen in Figure 5, it becomes obvious that the data set follows an exponential distribution. However, to be certain, a chi-square goodness-of-fit test is performed with a quantile-quantile plot to determine if the data does follow an exponential distribution.

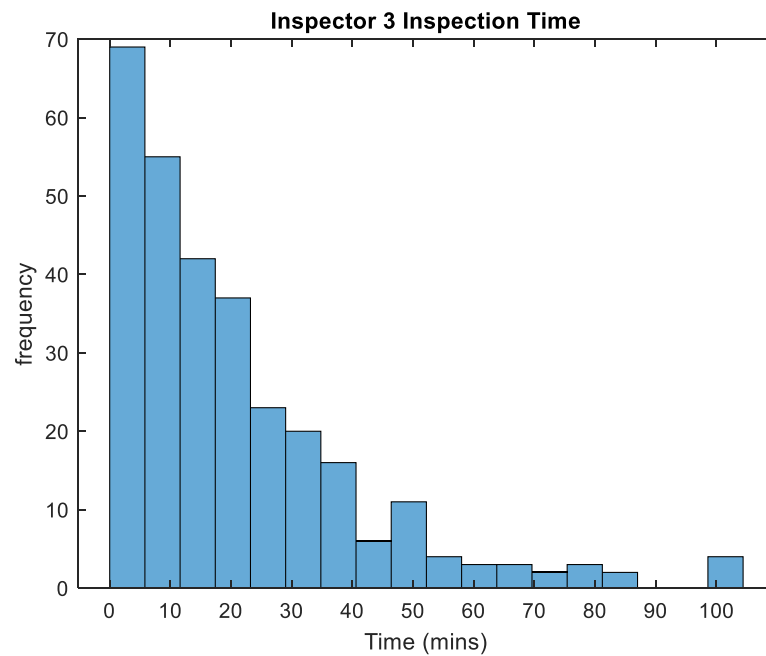


**Figure 6: Q-Q Plot of Inspector 2**

Here you can also observe (Figure 6) the plot confirms that the exponential distribution is a good fit with linearity of the graph. There are some outliers present in the 80-100<sup>th</sup> quantile, but the fit is still good. The parameter estimator used for this quantile-to-quantile graph is  $\lambda = \frac{1}{x}$  with a quantile function =  $X\ln(1 - x)$ .

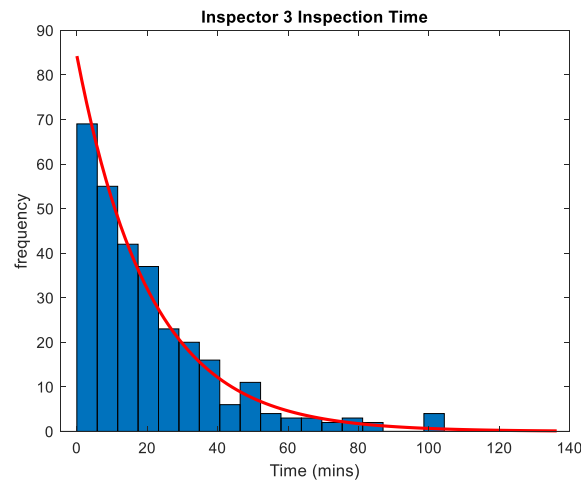
The chi-square goodness-of-fit test determines whether to reject or accept the null hypothesis that the data set follows an exponential distribution with an estimator of  $\lambda = 0.064362904$ . With a  $\alpha = 0.05$ , and with 16 degrees of freedom ( $18-1-1 = 16$ ) we obtain the critical value of 7.962. The chi-squared test statistic is 5.2890, therefore we do not reject the null hypothesis and we accept the data set does follow an exponential distribution at a significance of 5%.

### 5.3 Inspector 3



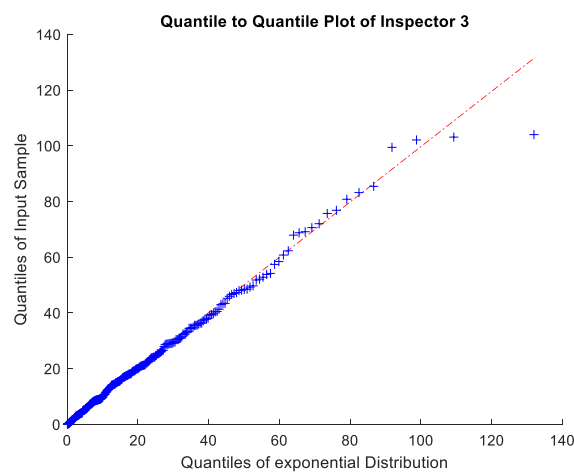
**Figure 7: Histogram of Inspector 3's Historical Service Times (18 bins)**

From the initial appearance of the histogram (Figure 7), it is clear the historical data set of Inspector 3 follows an exponential distribution.



**Figure 8: Inspector 3's Histogram with an Exponential Fit**

After overlaying an exponential function over the histogram (Figure 8) and it becomes obvious that the data set follows an exponential distribution. However, to be certain a chi-square goodness-of-fit test is performed with a quantile-quantile plot to determine if the data follow an exponential distribution.

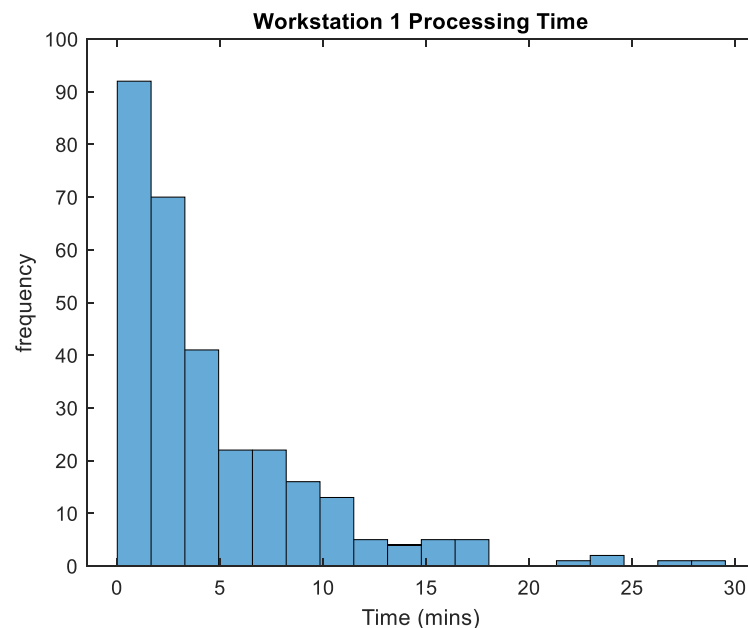


**Figure 9: Q-Q Plot of Inspector 2**

Here you can also observe (Figure 9) the plot confirms that the exponential distribution is a good fit because of the linearity of the graph. There are some outliers present in the 120-140<sup>th</sup> quantile, but the fit is still good. The parameter estimator used for this quantile-to-quantile graph is  $\lambda = \frac{1}{x}$  with a quantile function =  $X\ln(1 - x)$ .

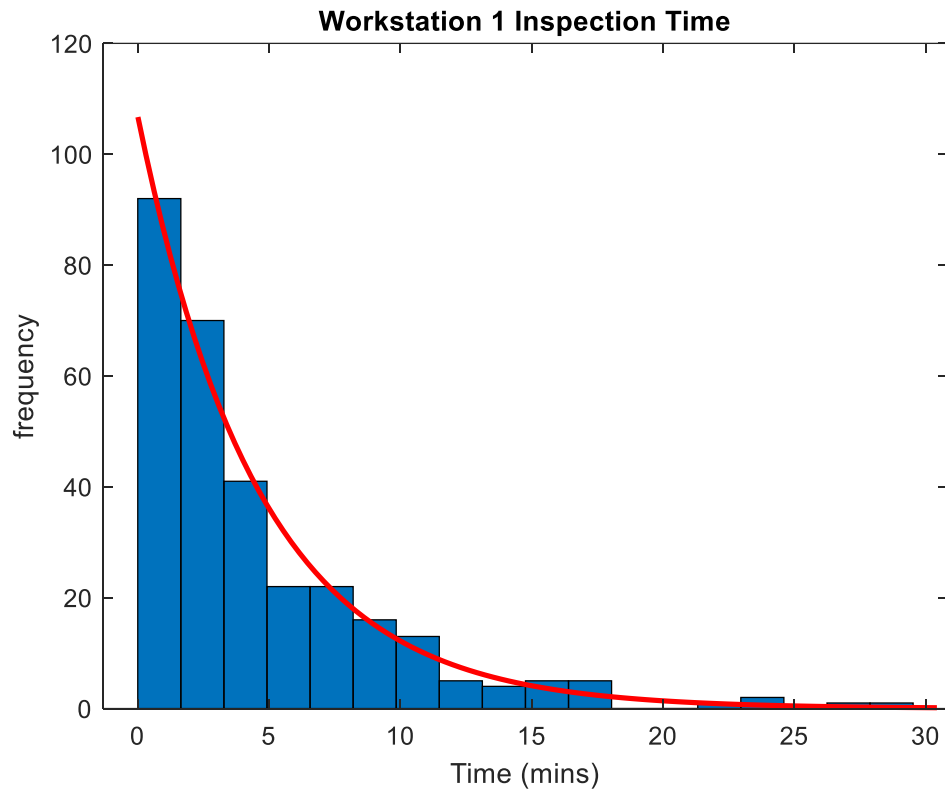
The chi-square goodness-of-fit test determines whether to reject or accept the null hypothesis that the data set follows an exponential distribution with an estimator of  $\lambda = 0.048466613$ . With a  $\alpha = 0.05$ , and with 16 degrees of freedom ( $18-1-1 = 16$ ) we obtain the critical value of 7.962. The chi-squared test statistic is 0.7268, therefore we do not reject the null hypothesis and we accept the data set does follow an exponential distribution at a significance of 5%.

## 5.4 Workstation 1



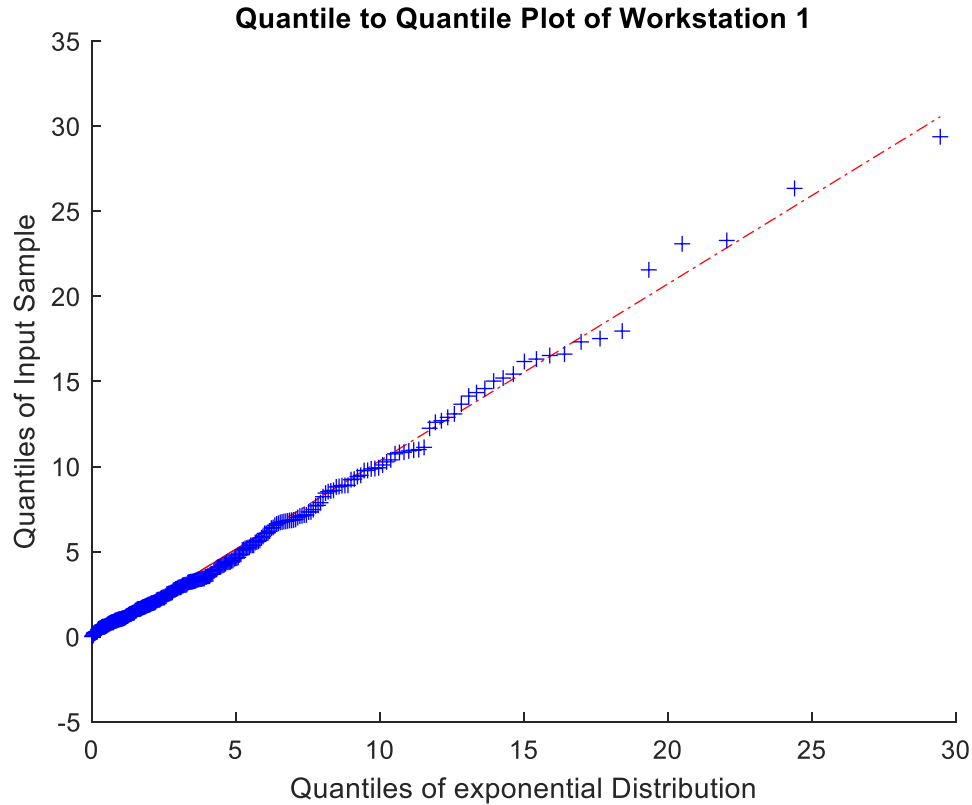
**Figure 10: Histogram of Workstation 1's Historical Processing Times (18 bins)**

From the initial appearance of the histogram (Figure 10), it is clear Workstation 1's historical data set follows an exponential distribution.



**Figure 11: Workstation 1's Histogram with an Exponential Fit**

After, overlaying an exponential function over the histogram (Figure 11) and it is even more obvious that the data set follows an exponential distribution. However, to be certain a chi-square goodness-of-fit test is performed with a quantile-quantile plot to determine if the data follow an exponential distribution.



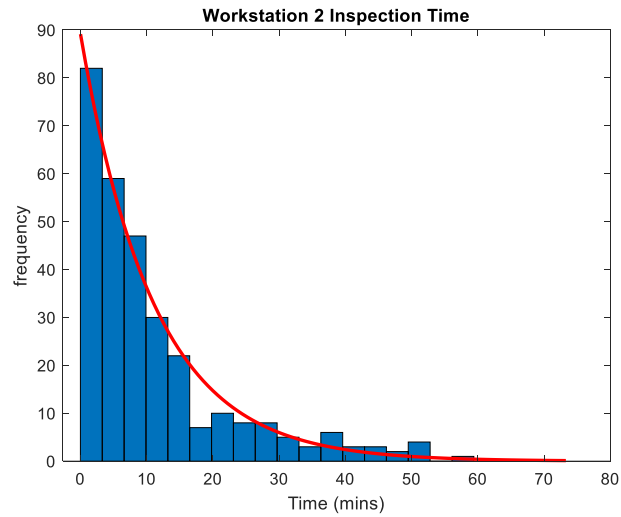
**Figure 12: Q-Q Plot of Workstation 1**

Here you can also observe (Figure 12) the plot confirms that the exponential distribution is a good fit because of the linearity of the graph. There are no outliers present in the quantile-to-quantile graph, meaning the fit is very good. The parameter estimator used for this quantile-to-quantile graph is  $\lambda = \frac{1}{x}$  with a quantile function  $= X \ln(1 - x)$ .

The chi-square goodness-of-fit test determines whether to reject or accept the null hypothesis that the data set follows an exponential distribution with an estimator of  $\lambda = 0.2171827$ . With a  $\alpha = 0.05$ , and with 16 degrees of freedom ( $18 - 1 - 1 = 16$ ) we obtain the critical value of 7.962. The chi-squared test statistic is 3.1573, therefore we do not reject the null hypothesis and we accept the data set does follow an exponential distribution at a significance of 5%.

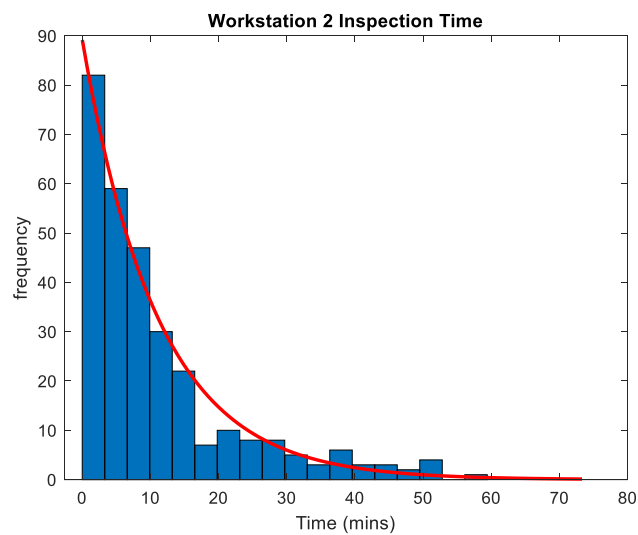


## 5.5 Workstation 2



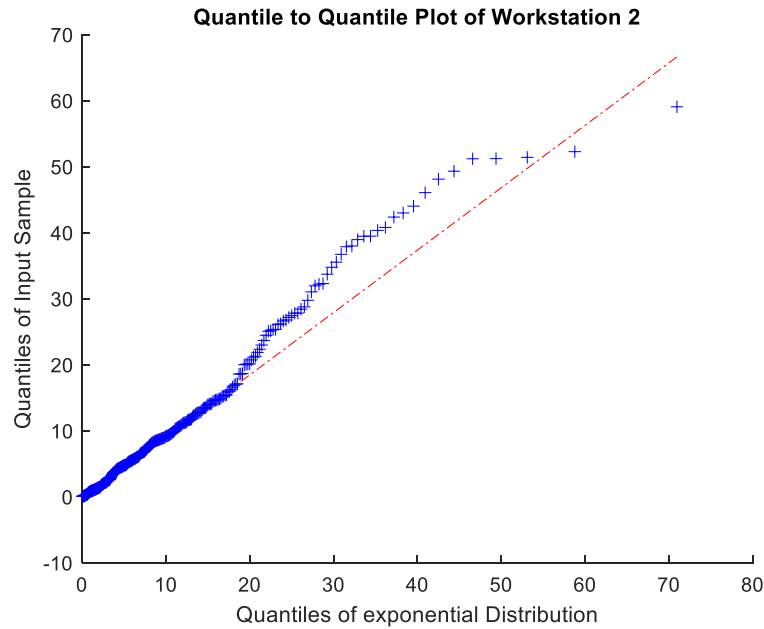
**Figure 13: Histogram of Workstation 2's Historical Processing Times (18 bins)**

From the initial appearance of the histogram (Figure 13), it appears Workstation 1's data set does follow an exponential distribution, but the bins from 20 onward deviate from the distribution somewhat.



**Figure 14: Workstation 2's Histogram with an Exponential Fit**

After overlaying an exponential function over the histogram (Figure 14) and the deviation observed at bins near 20 minutes is emphasized. It does appear the data set follows an exponential distribution; however, the histogram does deviate from the distribution and to be certain a chi-square goodness-of-fit test will be performed with a quantile-quantile plot to determine if the data follow an exponential distribution.

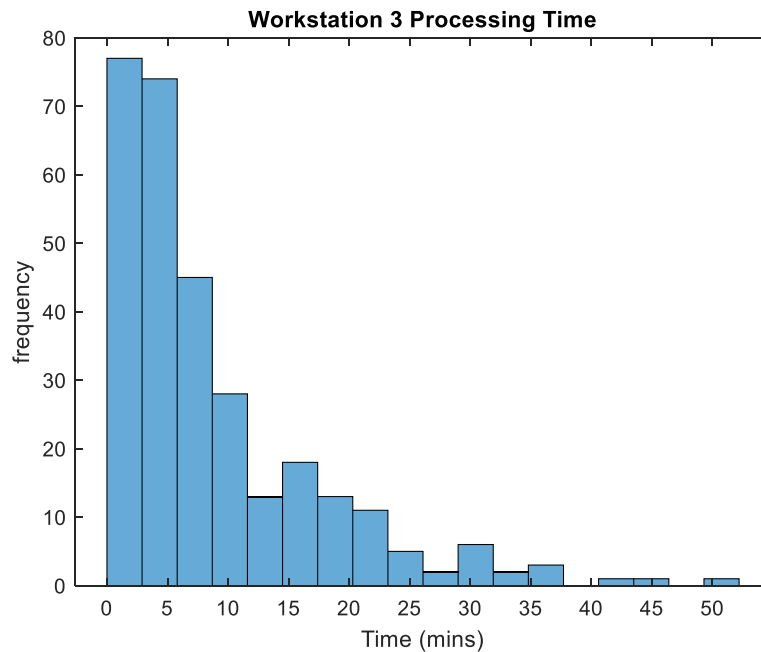


**Figure 15: Q-Q Plot of Workstation 2**

Here we observe (Figure 15) the plot confirms that the exponential distribution is NOT a good fit as the graph is non-linear. There are no outliers present in the quantile-to-quantile graph, this proves the data set truly is not representative of an exponential distribution. The parameter estimator used for this quantile-to-quantile graph is  $\lambda = \frac{1}{x}$  with a quantile function  $= X \ln(1 - x)$ .

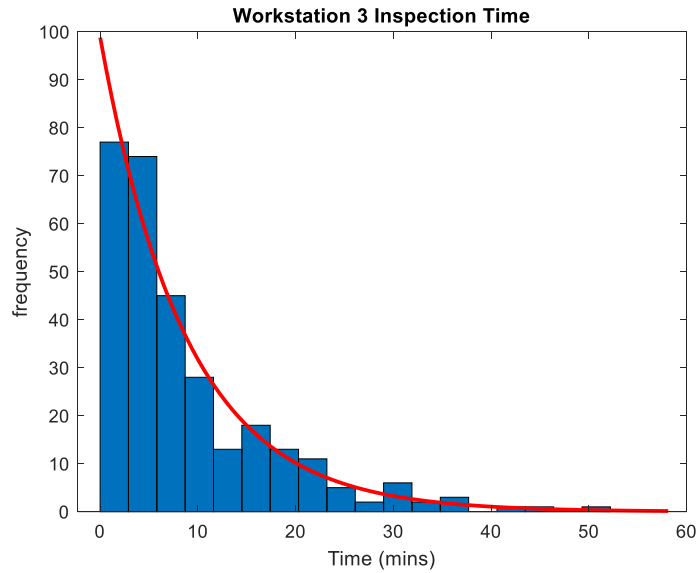
The chi-square goodness-of-fit test determines whether we reject or accept the null hypothesis that the data set follows an exponential distribution with an estimator of  $\lambda = 0.090150109$ . With a  $\alpha = 0.05$ , and with 16 degrees of freedom ( $18-1-1 = 16$ ) we obtain the critical value of 7.962. The chi-squared test statistic is 12.7743, therefore we reject the null hypothesis and we do not accept the data set does follow an exponential distribution at a significance of 5%.

## 5.6 Workstation 3



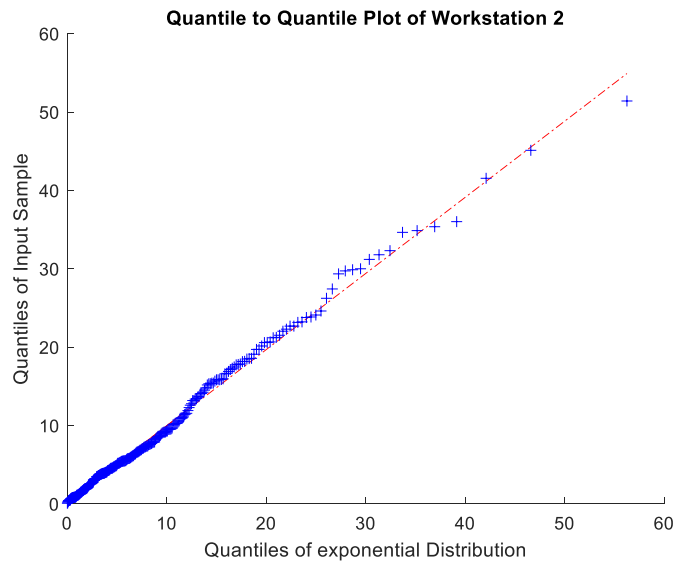
**Figure 16: Histogram of Workstation 3's Historical Processing Times (18 bins)**

From the initial appearance of the histogram (Figure 10), it looks like Workstation 3's data set follows an exponential distribution.



**Figure 17: Workstation 3's Histogram with an Exponential Fit**

After overlaying an exponential function over the histogram (Figure 17) and it becomes obvious that the data set follows an exponential distribution. However, to be certain a chi-square goodness-of-fit test is performed as well as a quantile-quantile plot to determine if the data follow an exponential distribution.



**Figure 18: Q-Q Plot of Workstation 3**

As you can also observe here (Figure 18) the plot confirms that the exponential distribution is a good fit because of the linearity of the graph. There no outliers present in the quantile-to-quantile graph, meaning the fit is very good. The parameter estimator used for this quantile-to-quantile graph is  $\lambda = \frac{1}{x}$  with a quantile function  $= X \ln(1 - x)$ .

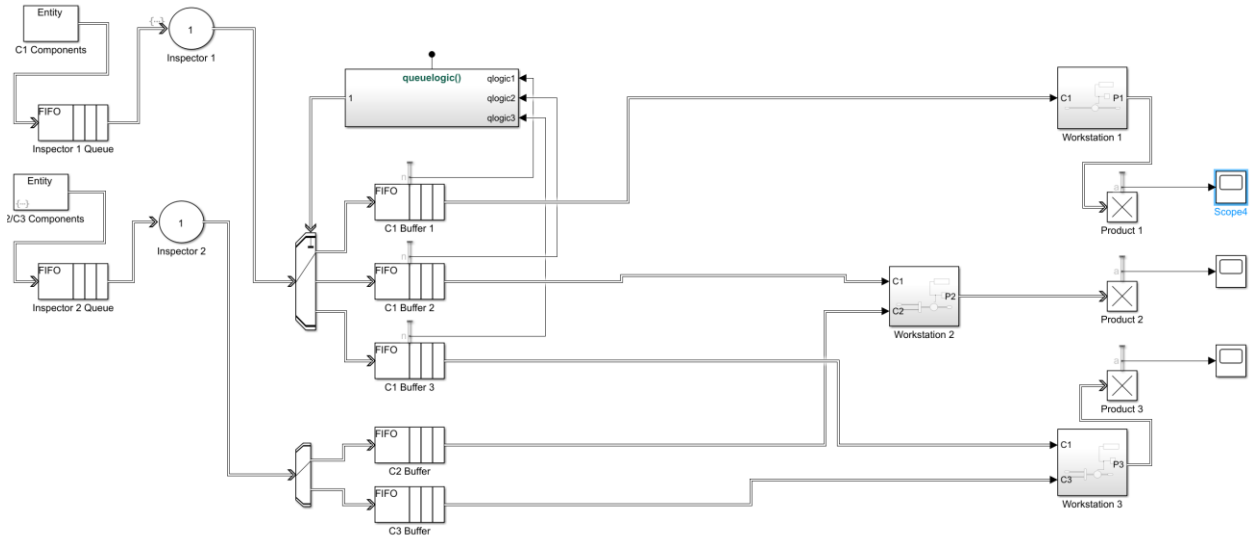
The chi-square goodness-of-fit test determines whether to reject or accept the null hypothesis that the data set follows an exponential distribution with an estimator of  $\lambda = 0.113693469$ . With a  $\alpha = 0.05$ , and with 16 degrees of freedom ( $18-1-1 = 16$ ) we obtain the critical value of 7.962. The chi-squared test statistic is 4.9434, therefore we do not reject the null hypothesis and we accept the data set does follow an exponential distribution at a significance of 5%.

## **5.7 Input Implementation**

Based on the histograms, the Q-Q plots, and the chi-square goodness-of-fit test, it was found that the inspector and workstation times could be modeled exponentially. One of the sets of data was not exponential, we can assume that case was an outlier. MATLAB has built in support for generating random numbers, `rand()` used based on a seed number. After the data files are read and the mean is calculated, a script utilizing `rand()` will generate a random number using a seed from each distribution.

## **6.0 MODEL VERIFICATION**

To verify this model, we have implemented several checks and balances to confirm that the outputs we determine are within reason. Each entity of the model is tested individually according to the strategies below. Essentially, we check the redundancy of the model.



## 6.1 Events

As previously discussed, the simulation should not process any events before the start time. To confirm this, the events' start time will be compared to the current clock time of the simulation. If a mismatch occurs, the simulation will stop and output the cause of the failure.

## 6.2 Inspectors

When the simulation starts, the inspectors should not be blocked and should be free to inspect and repair components for the workstations. During the initialization, our simulation will check to see if the inspectors are ready and not blocked from placing components into the buffers. If the inspectors cannot begin the inspection, the simulation will stop and output the cause of the failure.

## **6.3 Buffers**

When the simulation starts, buffer zones should not have anything in the queue and should be ready to receive components. During initialization, our simulation will check the buffer queues for components to determine if the model has been initialized correctly. If the buffer zones are full or have a component to start, the simulation will stop and output the cause of the failure.

### **6.3.1 Buffer Bounds**

As outlined in the model of the system, each workstation has a buffer capacity of two components. Our simulation will check to see if the buffer has two or less components after each event. If this is not the case, the simulation will stop and output the cause of the failure.

## **6.4 Workstations**

During the course of the simulation, the workstations should not be creating any products without the required components existing in the necessary buffer zones for each workstation. If this is the case, the simulation will stop and output the cause of the failure.

## **6.5 Products to Components Ratio**

At the end of the simulation, the number of components inspected should correspond to the number of products created. For example, if our simulation has more products produced than the expected number of components needed to create them, then we can assume that an error has occurred at some point during our simulation. At the end of the simulation, our script will verify that the ratio of products produced matches the components inspected.

## **7.0 MODEL VALIDATION**

To validate our model, we check our results from the simulation to determine if they are an accurate representation of what the real system would be.

### **7.1 Results Comparison**

Once our simulation has been completed, an output file containing the results will be compared to the expected values. The idle times, products produced, product distribution, workstation idle times and components inspected are checked to determine if they are reasonable for the simulation time. For example, if the inspector idle time is above the simulation time, then we cannot validate those results as acceptable. Also, idle times should match production, a workstation with little idle time should produce many products. For validation of the results, common sense is relied on to gauge how reasonable or unreasonable the results are.

### **7.2 Sensitivity Analysis**

Sensitivity analysis can be performed on the simulation times and random number generation. However, performing a sensitivity analysis on the random number seed is unreasonable as the results are unpredictable and cannot be analyzed for any meaningful conclusions.

For simulation times, lower values produce results that can vary significantly from each other. This is expected as shorter simulations do not have



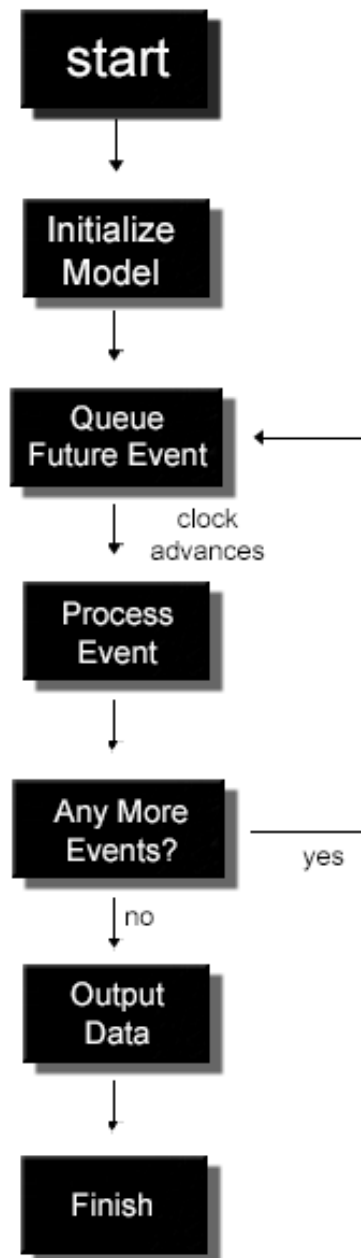
enough time to reach a steady state. Longer simulation times will hit a steady state and will not vary as much as the shorter ones. For this reason and based on the sensitivity analysis, we chose to run our simulation for a significant amount of time (24 hours/1440 minutes) so that simulation results do not vary substantially.

### **7.3 Historical Data as a Validation Alternative**

If this were a real system that we want to model and simulate, the system would have real-life results that we could compare to as an alternative to the validation methods above. Historical data is an infallible representation of how the system would perform in real life. As an alternative, we would compare the results of the simulation to those of the real-life system and confirm with certainty that the results we have achieved are accurate.

## 8.0 PRODUCTION RUNS AND ANALYSIS

Independent Replication Flowchart



To determine the quantities of interest, a seed of 652 was chosen to generate the random numbers. The simulation ran for 1440 minutes, and ten replications were done to make sure the confidence intervals had widths that did not exceed 20% of the estimated values. A flowchart was created to describe how each replication was performed.

To determine how many replications were required to achieve a width that did not exceed 20% of the estimated values, the confidence interval was calculated after each replication and if the confidence interval was too wide, an additional replication is performed until the width is 20% or less of the estimated values. Depending on the seed, the number of replications required can change, but for the seed chosen ten replications was enough.

It is important to note that for each replication the system will require some time to reach a steady

state. The system has a cold start where it requires some time to fill out buffers zones and ramp up production. After some analysis, the steady state of the system is around 230 minutes, which

is acceptable as the simulation time we chose was well above that number. To achieve distinct results in each replication, the variables are reset at the start of each replication.

The replication results are posted below, replications are performed with default behaviour indicated for each inspector and workstation in the project description.

	Mean	Variance	Confidence Interval (alpha = 0.05)
<b># of Component 1 Inspected</b>	112.9	46.76	[108.30-117.49]
<b># of Component 2 Inspected</b>	10.70	9.12	[8.67-12.72]
<b># of Component 3 Inspected</b>	2.10	1.43	[1.29-2.90]
<b># of Product 1 Produced</b>	99.60	44.04	[95.14-104.05]
<b># of Product 2 Produced</b>	11.20	6.62	[9.47-12.92]
<b># of Product 3 Produced</b>	2.10	1.43	[1.29-2.90]
<b>Inspector 1 Idle % (0 to 1)</b>	0.00	0.00	[0-0]
<b>Inspector 2 Idle % (0 to 1)</b>	0.78	0.00094	[0.76-0.80]
<b>Workstation 1 % (0 to 1)</b>	0.52	0.0012	[0.50-0.54]
<b>Workstation 2 % (0 to 1)</b>	0.76	0.00086	[0.74-0.78]
<b>Workstation 3 % (0 to 1)</b>	0.82	0.00019	[0.81-0.83]

Just as we hypothesized in Milestone 1, Workstation 1 has the highest rate of production. Product 1 only requires one component and has a singular Inspector who prioritizes providing the component to Workstation 1. Product 2 and Product 3 require the services of both Inspectors, need two components to be completed, and require a shared component with Product 1. With the

default priority defined in the project description, the inspectors supply to the Workstation with the shortest queue and prioritize Workstations 1, 2, 3 respectively in the event of a tie. This means that most Component 1s are transferred to Workstation 1's buffer zone as it is producing the fastest and will usually have the shortest queue. This is evident by Workstation 1's idle time of 52%, which is the lowest of the three workstations. Inspector 1 was idle for 0% of the simulation time due to how fast Product 1 was being produced. This is due to its default priority configuration and singular component cost. 88% of all Component 1s were provided to Workstation 1. Ideally, this is not acceptable as Product 1 is produced ten times more than Product 2 and 3 combined. Inspector 2 had no choice but to wait, as all Component 2 and 3 buffer zones were max capacity. The production of workstations 2 and 3 was limited by the amount of Component 1s given. This is evident by the 1 to 1 ratio of Component 1s given to Workstation 2 and 3 and the number of Product 2 and 3s produced. This system is not very efficient as Inspector 2, Workstation 2, and Workstation 3 are idle on average for 78% of the simulation run. Too much priority is placed on creating Product 1. If this is the desired result, then this system is working as intended. If not, a significant production bias is placed on Product 1 and wages are wasted on Inspector 2 as well as the operating costs for Workstation 2 and 3. For there to be only two Product 3s produced in twenty-four hours is flat out unacceptable. The difference between Product 2 and 3 can be attributed to the higher priority of Workstation 2 over 3. However, an idle percentage of 52% is also not particularly good for Workstation 1, it is clear there needs to be additional Inspectors to repair and clean Component 1.

## **9.0 ALTERNATIVE OPERATING POLICY**

Due to the extreme idle time of both inspectors and all workstations, an alternative operating policy is necessary as the productivity of the system is quite subpar. As mentioned before, there are two changes that are very obvious to make, both happen to concern the default behaviours of the inspectors. Firstly, Inspector 1 chooses to route Component 1's to the shortest queue, as we determined previously, this method of routing tends to prioritize Workstation 1 and Product 1. Instead as an alternative design, Inspector 1 implements a round-robin method of distributing Component 1. Each Workstation will get Component 1's in turn, starting from Component 1. The other behaviour we target is how Inspector 1 responds to ties, previously the inspector prioritizes Workstation 1, Workstation 2 and 3 respectively regarding Component 1 distribution. This is changed to the opposite where the inspector prioritizes Workstation 3, Workstation 2 and 1 respectively regarding Component 1 distribution. These alternative designs will be tested and examined to determine the efficacy of the changes to the model. The goal of this operating policy change is to facilitate the increased production of Product 2 and 3 and an even ratio between products.

## 9.1 Alternative Component 2 Routing

Again, to determine the efficacy of the changes, the seed 652 was chosen and the simulation ran for 1440 minutes. The alternative design was replicated ten times. The simulation was changed to reflect the new round-robin method of distributing Component 1s.

	Mean	Variance	Confidence Interval (alpha = 0.05)
# of Component 1 Inspected	112.7	33.12	[108.58-116.81]
# of Component 2 Inspected	31.00	18.22	[27.94- 34.05]
# of Component 3 Inspected	31.10	20.99	[27.82-34.37]

<b># of Product 1 Produced</b>	50.20	63.95	[44.47-55.92]
<b># of Product 2 Produced</b>	31.30	21.56	[27.97-34.62]
<b># of Product 3 Produced</b>	30.90	22.76	[27.48-34.31]
<b>Inspector 1 Idle % (0 to 1)</b>	0.004379	0.000065	[0.00-0.01]
<b>Inspector 2 Idle % (0 to 1)</b>	0.063569	0.003665	[0.02-0.107]
<b>Workstation 1 % (0 to 1)</b>	0.67	0.0013	[0.65-0.70]
<b>Workstation 2 % (0 to 1)</b>	0.61	0.0074	[0.54-0.67]
<b>Workstation 3 % (0 to 1)</b>	0.65	0.0013	[0.62-0.67]

Here we can see that Inspector 2 actually has a job to do, previously Inspector 2 was idle for 78% of the time, by changing the Component 1 distribution to a round robin method Inspector 2's idle time dropped to 0.004%. No longer was Inspector 2 blocked from processing Component 2 and 3 because the workstations were waiting for Component 1. In this alternative design, Component 1 is distributed relatively evenly with respect to Product 1's quick production time. With these changes, more Product 2 and 3s were created than Product 1, but more Product 1 was created than each product individually. Compared to the default behaviour, the workstations are still idle 64% of the time, this percentage is still quite high. In terms of the whole system's efficiency, the total number of products produced was still the same at 112. The number of products produced as a whole remains relative to the number of Component 1s inspected. In the default configuration, Inspector 1 was never idle and inspected as many Component 1s as possible. It seems the true bottleneck lies in the limited number of Inspectors on Component 1. The number of Inspectors on Component 1 needs to match the workstations throughput to maximize the systems potential. In terms of a desired result, this change only

created a more even ratio between the products but did not improve the systems efficiency. It is clear the number of inspectors in the system should match the need and demand of Component 1s. By adding an additional inspector, the total number products would increase significantly, whilst having inspector still utilized fully.

## 9.2 Alternative Component 1 Priority

To test the alternative design, the simulation ran for 1440 minutes with the seed 652 to generate the random numbers. The alternative design choices were replicated ten times. The model and simulation have been changed to reflect the new ranking when distributing Component 1 in the event of a tie.

	Mean	Variance	Confidence Interval (alpha = 0.05)
# of Component 1 Inspected	112.9	46.76	[108.00-117.79]
# of Component 2 Inspected	28.30	9.78	[26.06-30.53]
# of Component 3 Inspected	33.50	36.05	[29.20-37.79]
# of Product 1 Produced	49.60	70.93	[43.57-55.62]
# of Product 2 Produced	28.80	10.17	[26.51-31.08]
# of Product 3 Produced	34.20	33.95	[30.03-38.36]
Inspector 1 Idle % (0 to 1)	0.0012	0.000007	[0-0.0031]
Inspector 2 Idle % (0 to 1)	0.044735	0.0013	[0.018-0.071]
Workstation 1 % (0 to 1)	0.68	0.001395	[0.65-0.70]
Workstation 2 % (0 to 1)	0.63	0.003527	[0.58-0.67]
Workstation 3 % (0 to 1)	0.62	0.003680	[0.57-0.66]

Just as we feared, changing the behaviours of the inspectors did not influence the total system output. Again, the average idle time of the workstations was still 64%, it is obvious that the system is not creating products at its full potential. In comparison to the default behaviour both the inspectors are working on average 99.7% of the time, with this behaviour change both inspectors are doing as work as they can. Compared to the alternative routing method, Product 3 was produced more than Product 2 because of the priority change in the event of a tie. Again, the total number of products produced is tied to the number of Component 1s inspected and remains at 112 total products average. Neither of these alternative operating policies can be recommended, as the total system output remains the same. However, if a more even ratio between products is desired, this is the way to go.

## **10.0 CONCLUSION**

After running the simulation based off our model, we determined that the current system is limited by the number of Component 1s inspected. No matter how the behaviour of the inspectors changed, the total number of products made stayed the same. It was determined the number of inspectors on Component 1 was not sufficient to supply each workstation within a timely manner. The number of Component 1s inspected equalled the number products produced, if Inspector 1 was always busy, nothing would change. However, product ratios could be manipulated based on inspector behaviour. By default, the number of Product 1s dominated the number of other products made. In this scenario, Inspector 1 is never idle while Inspector 2 is constantly blocked from inspecting Component 2 and 3s. Inspector 2 was idle 78% of the time with the default behaviour, a staggering 18.72 hours out of 24. The default behaviour does not



maximize the work of each inspector, but still maximized the total number of products. It is heavily recommended to improve the performance of the system, additional inspectors are required on Component 1.

	<b>Changed Behaviour</b>	<b>Maximum Output</b>	<b>% Increase</b>
<b># of Product 1 Produced</b>	49.60	72.94	147% increase
<b># of Product 2 Produced</b>	28.80	45.71	158% increase
<b># of Product 3 Produced</b>	34.20	55.16	161% increase

If the system had additional inspectors to maximize the workstations output, on average each product made would see an increase of about 155%, while the total number of products increase by 153%. Additional inspectors beyond this threshold, will require additional workstations to maximize their potential and vice versa. In conclusion, this system is not at its maximum potential, it heavily favors Product 1 and can easily be changed to improve its performance. If Product 1 is desired, additional inspectors with default behaviour is a great choice. If an even ratio of products is required, then additional inspectors on Component with a round-robin even distribution is ideal.