Project Report

Name: Chintan Jayeshkumar Kharadi

Student Id: 101169771

Course Name: Discrete Simulation/Modelling

Couse Code: SYSC 5001

Professor: C.Huang

Table of contents Page Number

Deliverable 1:

System Description:

A Manufacturing facility assembles three products P1, P2 and P3 in three workstations W1, W2 and W3. Product P1 consists of C1, P2 consists of C1 and C2 and P3 consists of C1 and C3. Here C1, C2 and C3 are components. These components are cleaned and repaired by inspectors. Inspector 1 handles component 1 and Inspector 2 handles component 2 and 3.

Problem Formulation, Setting of Objectives and Overall Project Plan:

Iteration	Problem Formulation	Setting of Objectives and Overall Project Plan
1	Manufacturing facility assembles products. Inspector does the inspection and passes the components to the respective workstation. Each station has buffer capacity of 2. C1 is used for product 1.	How to calculate the average processing time for workstation 1 and inspection time of inspector 1? Criterion for evaluating the effectiveness: 1) 3 people for 5 days for data collection 2) 1 person for 3 days for data analysis 3) 1 person for 4 days for model building 4) 1 person for 3 days for running the model 5) 1 person for 2 days for model implementation
2	Manufacturing facility assembles products. Inspector 2 does the inspection of components 2 and 3 and passes the components to the respective workstation. Each station has buffer capacity of 2. C1 and C2 are used for product 2.	How to calculate the average processing time for workstation 2 and inspection time of inspector 2? Criterion for evaluating the effectiveness: 1) 3 people for 5 days for data collection 2) 2 person for 3 days for data analysis 3) 1 person for 4 days for model building 4) 2 person for 3 days for running the model 5) 1 person for 2 days for model implementation

3	Manufacturing facility assembles products. Inspector 2 does the inspection of components 2 and 3 and passes the components to the respective workstation. Each station has buffer capacity of 2. C3 and C1 are used for product 3.	How to calculate the average processing time for workstation 3 and inspection time of inspector 2? Criterion for evaluating the effectiveness: 1) 3 people for 5 days for data collection 2) 2 person for 3 days for data analysis 3) 1 person for 4 days for model building 4) 2 person for 3 days for running the model 5) 1 person for 2 days for model implementation
4	Same as above plus the following: Component 1 is required at 3 workstations. Workstation require component 1 frequently. Workstation 1 has higher priority than other workstation but workstation 2 and 3 are waiting for longer time.	What should we do to increase the inspection of component 1? What is the average time of inspection of component 1? Criterion for evaluating the effectiveness: 1) 4 people for 7 days for data collection 2) 3 person for 5 days for data analysis 3) 2 person for 6 days for model building 4) 3 person for 5 days for running the model 5) 2 person for 3 days for model implementation
5	Same as above plus the following: Buffer is full for any of the three workstation. Respective Inspector is blocked until there is an opening. For example buffer for workstation 1 is full then Inspector 1 is blocked.	How can we reduce the processing time to make products at workstation? What is the average time(mean) of each product or components at respective workstation ? Criterion for evaluating the effectiveness: 1) 5 people for 10 days for data collection 2) 4 person for 6 days for data analysis 3) 2 person for 4 days for model building

		4) 4 person for 6 days for running the model 5) 2 person for 3 days for model implementation

Model Conceptualization:

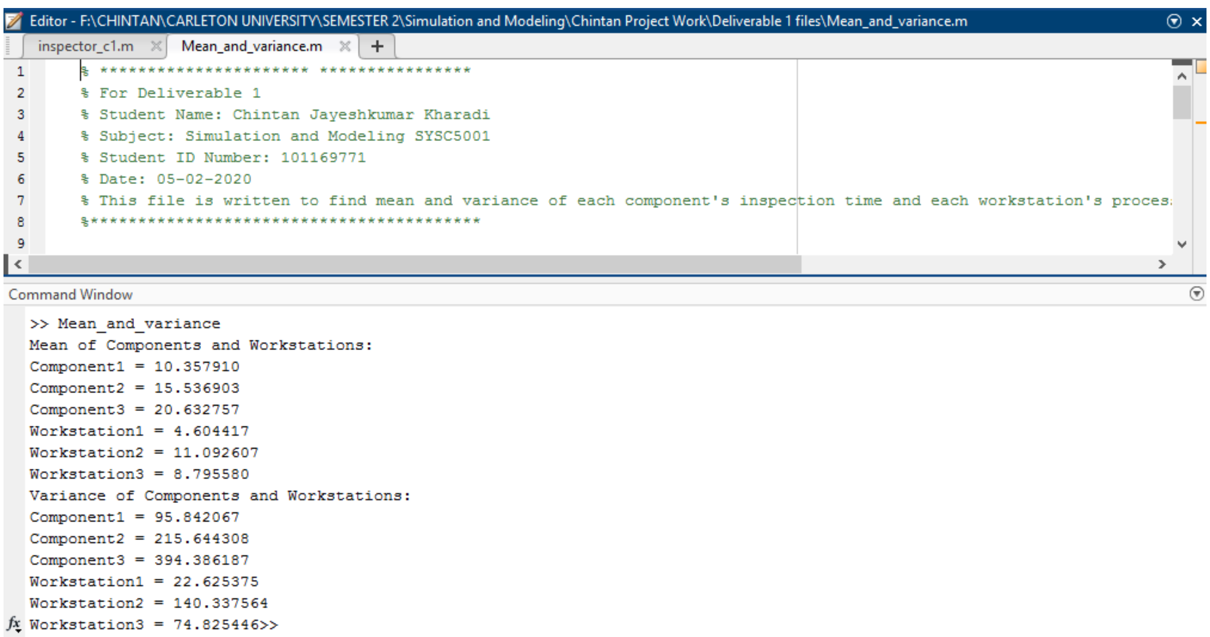
Entities and their Interactions:

Here, Entities are components and products.

Interactions are inspecting the components, delivering the components to workstations, assembling the components to make products.

Essential Features of the model:

- Component 1 is the only component which is inspected by inspector 1. Mean time for component 1 is high (10.357910) compared to the workstation 1 processing time(4.604417). This shows that workstation 1 will have to wait for many times as this component is used by all three workstations.



```
Editor - F:\CHINTAN\CARLETON UNIVERSITY\SEMESTER 2\Simulation and Modeling\Chintan Project Work\Deliverable 1 files\Mean_and_variance.m
inspector_c1.m   Mean_and_variance.m
% ********************** ****************
% For Deliverable 1
% Student Name: Chintan Jayeshkumar Kharadi
% Subject: Simulation and Modeling SYSC5001
% Student ID Number: 101169771
% Date: 05-02-2020
% This file is written to find mean and variance of each component's inspection time and each workstation's proces
%*****************************************
```

```
Command Window
>> Mean_and_variance
Mean of Components and Workstations:
Component1 = 10.357910
Component2 = 15.536903
Component3 = 20.632757
Workstation1 = 4.604417
Workstation2 = 11.092607
Workstation3 = 8.795580
Variance of Components and Workstations:
Component1 = 95.842067
Component2 = 215.644308
Component3 = 394.386187
Workstation1 = 22.625375
Workstation2 = 140.337564
Workstation3 = 74.825446>>
```

Figure 1 Mean and Variance

- Same statement is applied for mean processing time of workstation 3 and inspection time of component 3. Inspector 2 takes 20.63s of mean time to inspect component 3 while workstation 3 assembles and produces product in 8.79s of mean time. So, workstation 3 might have to wait for component 3.

Assumptions:

For Simple Model:

1) Components are infinite and they are always available to use.
2) Routing policy is available to route the components to the shortest queue.
3) For components 1 to be transferred to all three workstation, priority is not given and routing is given as W1-W2-W3-W1.

For Complex Model:

1) If buffers are full for component 1, then highest priority will be given to workstation 1 and then least priority is given to workstation 3. If the simulation doesn't give products 3 then change the priority accordingly.

Model Translation:

Matlab is used to program and simulate the model. The primary reason to select this is because of its user interface, APIs and libraries. It is well known for doing simulation, data analysis, visualization, algorithm development. The secondary reason is I am aware of Matlab and used it in projects. So, I am able to implement basic code with ease and it won't take much time to understand and learn new libraries and implement new toolbox in this project.

Model Implementation with Detailed level code:

Below is the diagram of high level implementation with software architecture.

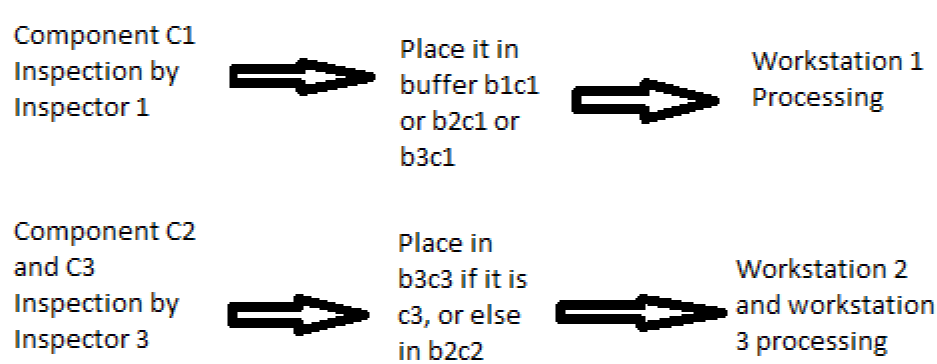


Figure 2 HIgh Level Implementation of Model

- Component C1 Inspection by Inspector 1: This module inspects C1 components and the inspection time will be recorded in this module and it will be sent to Buffer b1c1 or b2c1 orb3c1 according to priority and buffer availability. This module is implemented in Main_simulation.m and Inspector1.m . If all the buffers are loaded with components, then inspector is blocked, so idle time of inspector 1 is also calculated in Inspector1.m file.

- Place it in bufferb1c1 or b2c1 or b3c1: The logic of putting c1 component in the particular buffer is written in Inspector1.m file.
- Workstation 1 Processing: The Workstation 1 will check if buffer b1c1 is empty or not. If it is empty then it will record idle time of workstation 1. If it is not empty, then the service time of component 1 will be calculated in Main_simulation.m file. This will be optimized in workstation1.m file.
- Component C2 and C3 inspection by Inspector 2: This module first randomly assign priority to C2 or C3 component. Inspector 2 will get C3 or C3 randomly and will inspect accordingly. Inspector 2 will inspect the components and send it to corresponding buffers. The code for inspection of components is written in inspector2.m file and Random assignment of C2 or C3 is coded in Main_simulation.m file.
- Place in b3c3 if it is C3 or else in b2c2: This logic is implemented in Inspector2.m file. It will decide which buffer it will select and place it in buffer and count the inspection time. If the buffers are full, then it will count idle time of inspector 2.
- Workstation 2 and Workstation 3 processing: Workstation 2 and workstation 3 processing time is calculated in main_simulation.m file and this will optimized and separated in workstation2.m and workstation3.m file accordingly.

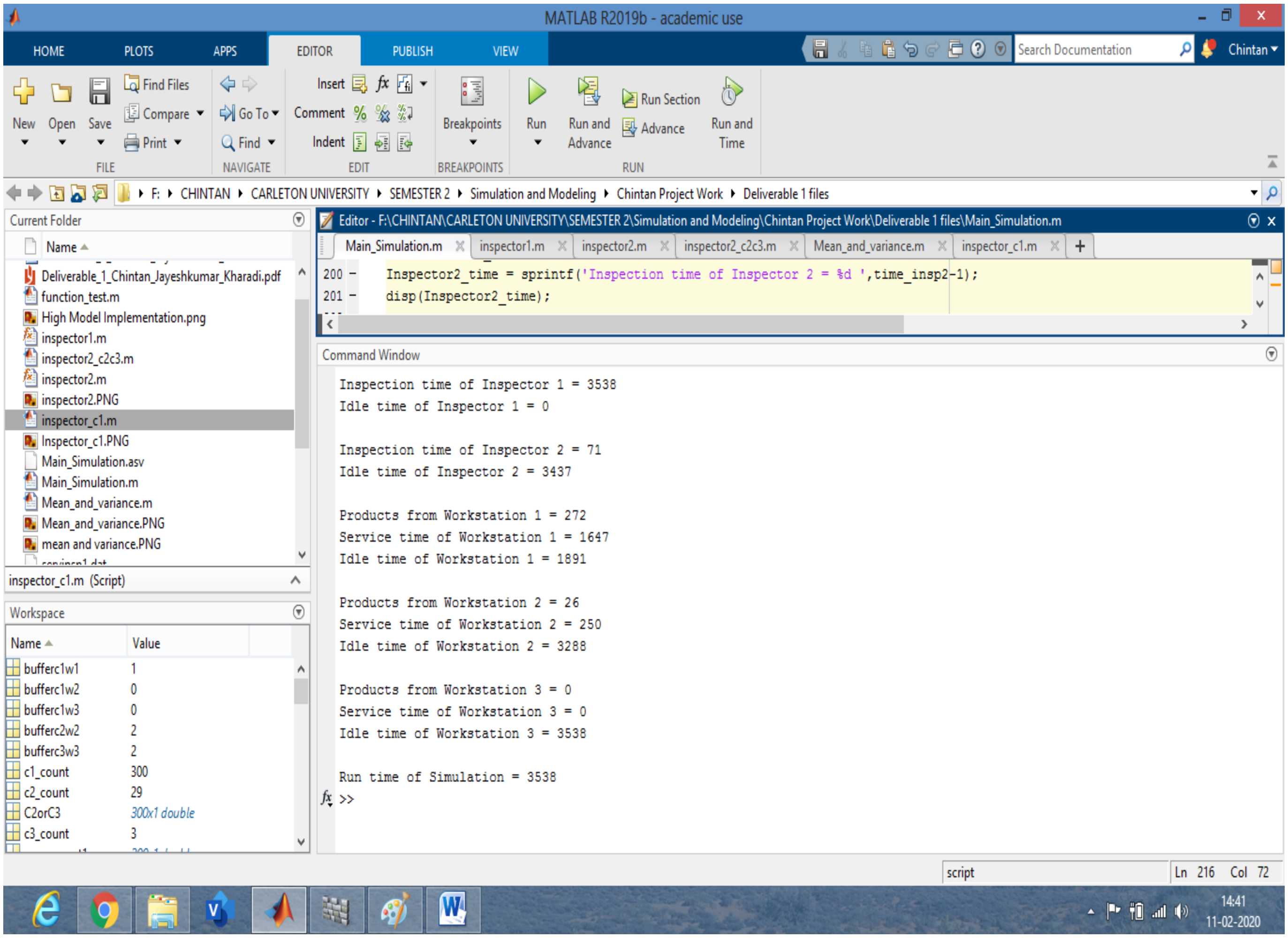
Detailed level code and It's Output:

Files created for the simulation:

1. Main_Simulation.m : This will simulate the whole manufacturing facility according to the logic given in project description.
2. Inspector1.m : This is a file that is used as a function in Main_simulation.m . It will give inspection time or idle time of inspector 1 and will decide the buffer logic.
3. Inspector2.m: This is a file that is used as a function in Main_simulation.m. It will generate inspection time or idle time of inspector 2 and will decide the buffer logic.
4. Inspector2_c2c3: This file is written to find Normal and exponential distribution of Inspector 2's inspection time
5. Inspector_c1.m: This file is written to find Normal and exponential distribution of Inspector 1's inspection time
6. MeanandVariance.m: This file is written to find mean and variance of each component's inspection time and each workstation's processing time

Simuation output:

All the source code files are attached with report(.m files) and the output of Main_Simulation is given as below:



I also implemented normal and exponential distribution for inspector 1 and inspector 2's inspection time for component 1 ,2 and 3. The code and output is displayed below:

Inspection time for component C1 by Inspector 1 and its Exponential and Normal Distribution:

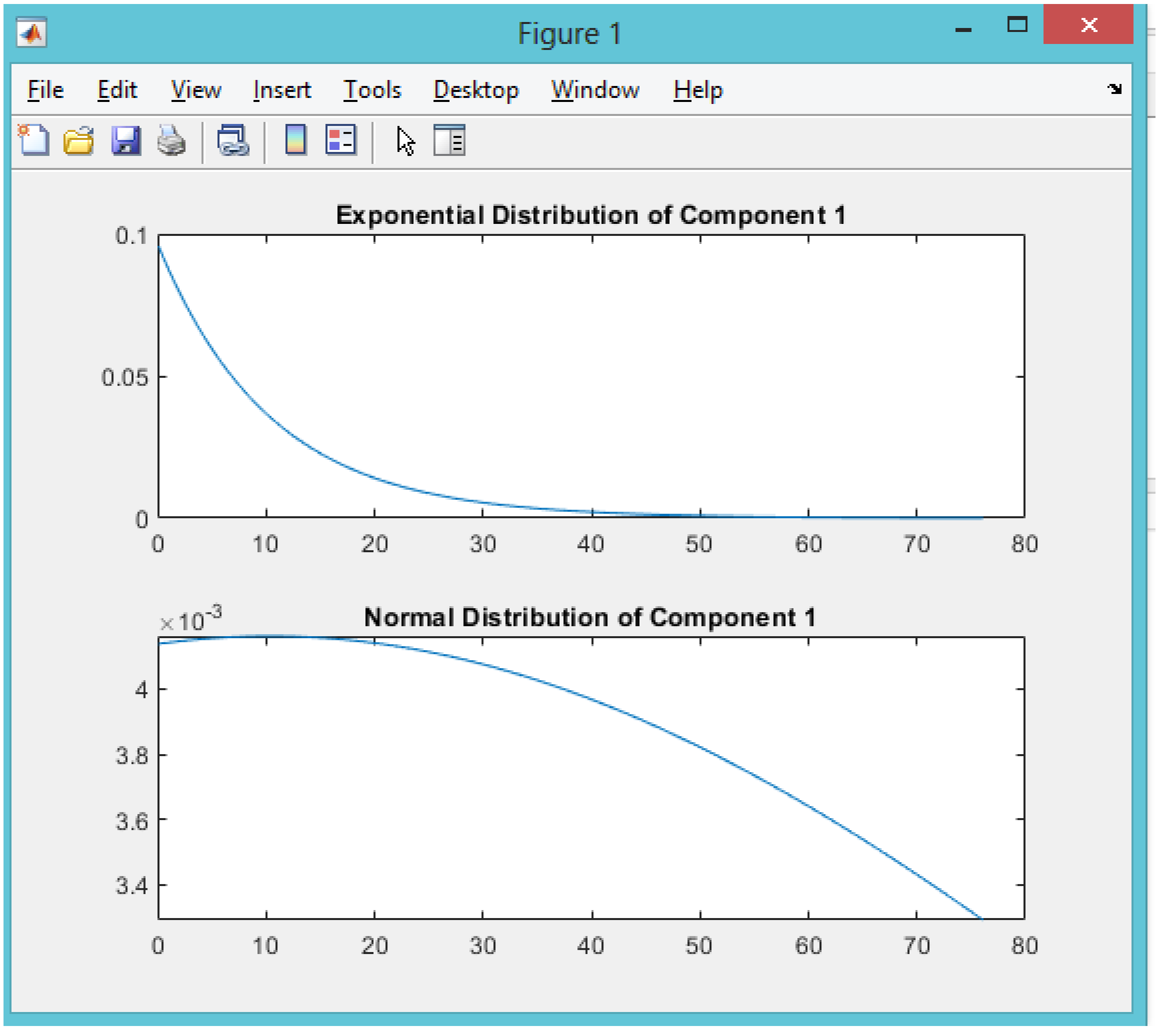
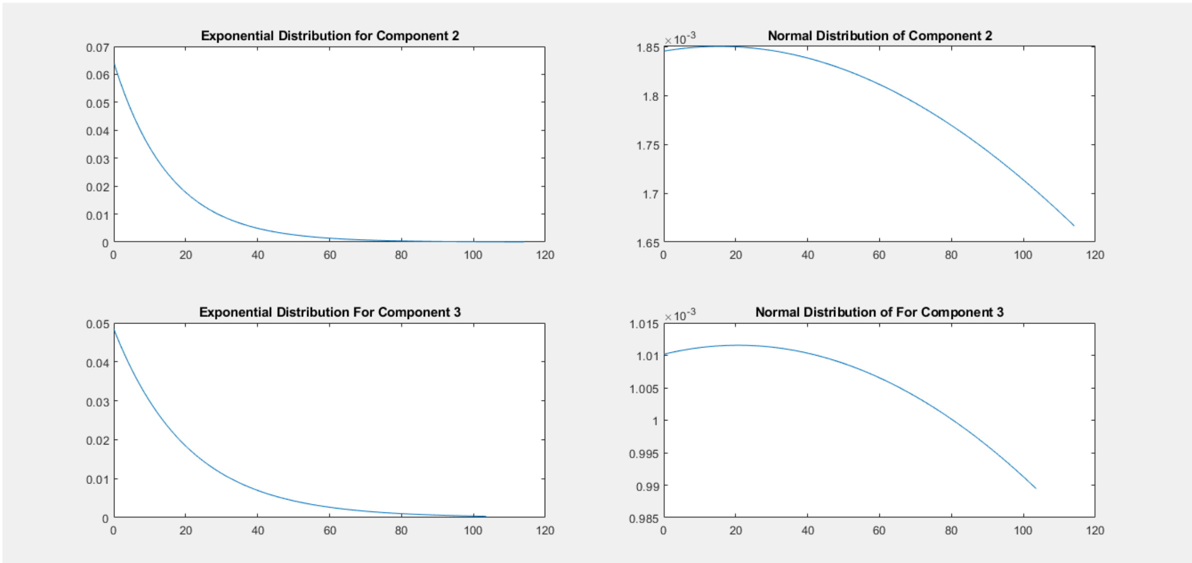


Figure 3 Normal And Exponential Distribution of Component 1

Inspection time for component C2 and C3 by Inspector 2 and their Exponential and Normal Distribution:



Mean and Variance of each component and workstation:

Command Window

```
>> Mean_and_variance
Mean of Components and Workstations:
Component1 = 10.357910
Component2 = 15.536903
Component3 = 20.632757
Workstation1 = 4.604417
Workstation2 = 11.092607
Workstation3 = 8.795580
Variance of Components and Workstations:
Component1 = 95.842067
Component2 = 215.644308
Component3 = 394.386187
Workstation1 = 22.625375
Workstation2 = 140.337564
Workstation3 = 74.825446>>
```

Deliverable 2:

Project Description:

In this deliverable, I have first used the random data that is given by professor for inspection time and service time of inspector 1, 2 and workstation 1,2,3 respectively. Given data is discrete.

The distribution of each set of data is identified by the histograms. Histogram is useful for determining the shape of a distribution. The number of class interval is based on the respective data.

By looking at the histogram, it is easier to identify that it follows exponential and normal and uniform distribution. But it is not easy to identify that it follows Weibull distribution. In this deliverable, I have identified 3 distribution which are normal, Weibull and exponential. The distribution here is not uniform as the inspection time and service time varies frequently. Each set of data distribution fit is now identified with Q-Q plots.

Goodness of fit test is usually for conducting null hypothesis on input data distribution . Here, I have used Chi square test for goodness of fit test.

The output for distribution of inspection and service time random variable data set is given as below:

Inspector 1's Component 1 Distribution Output :

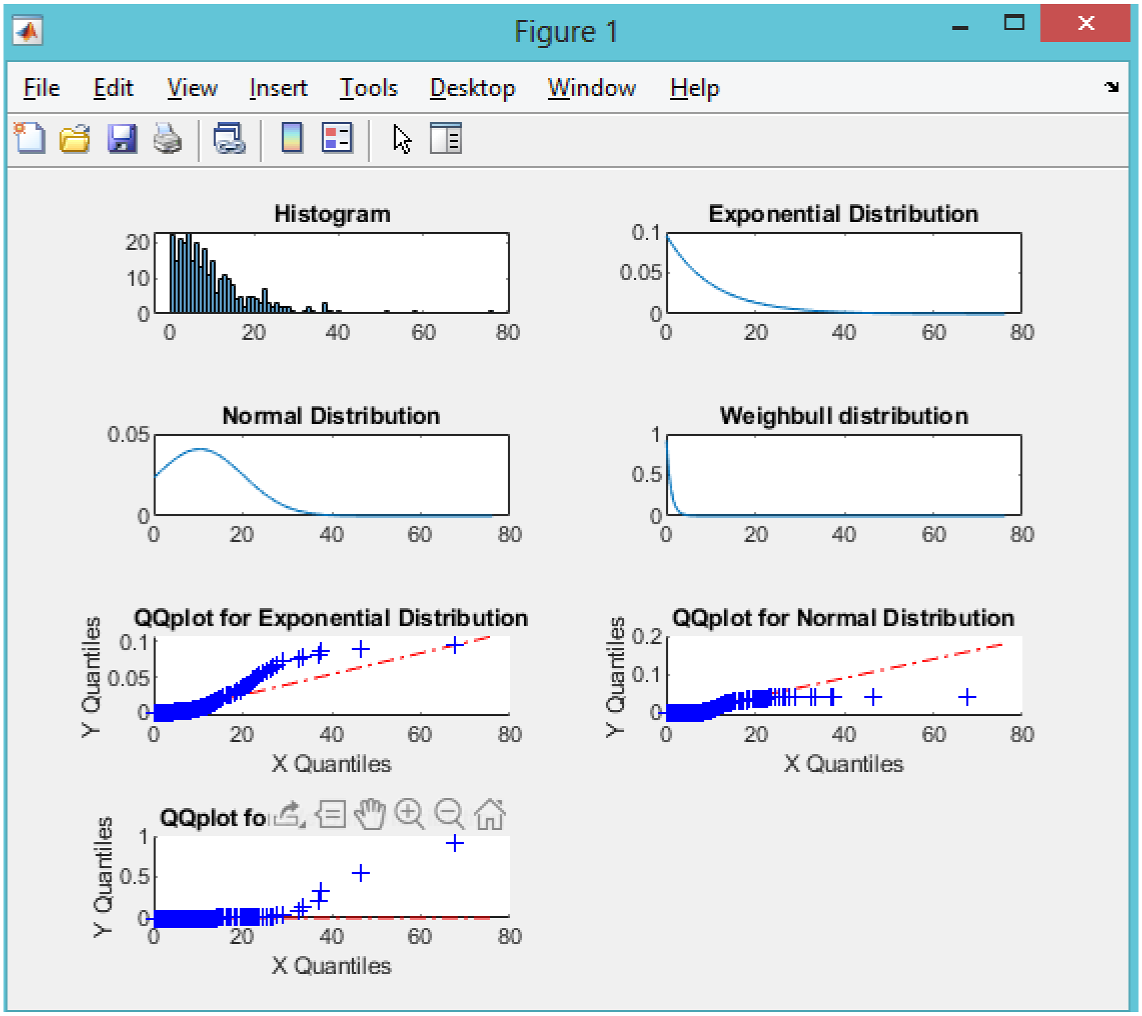


Figure 4Component 1 inspection time distribution

```
Command Window
Chi Square Test:
Null hypothesis for Exponential Distribution is accepted at 0.05 significance level
Null hypothesis for Normal Distribution is rejected at 0.05 significance level
Null hypothesis for Weibull Distribution is accepted at 0.05 significance level
>>
```

Figure 5 Null hypothesis for inspection time

It is clear that exponential and Weibull distribution is accepted as it follows this distribution but normal distribution is rejected.

Distribution of Inspection time of component 2 Output:

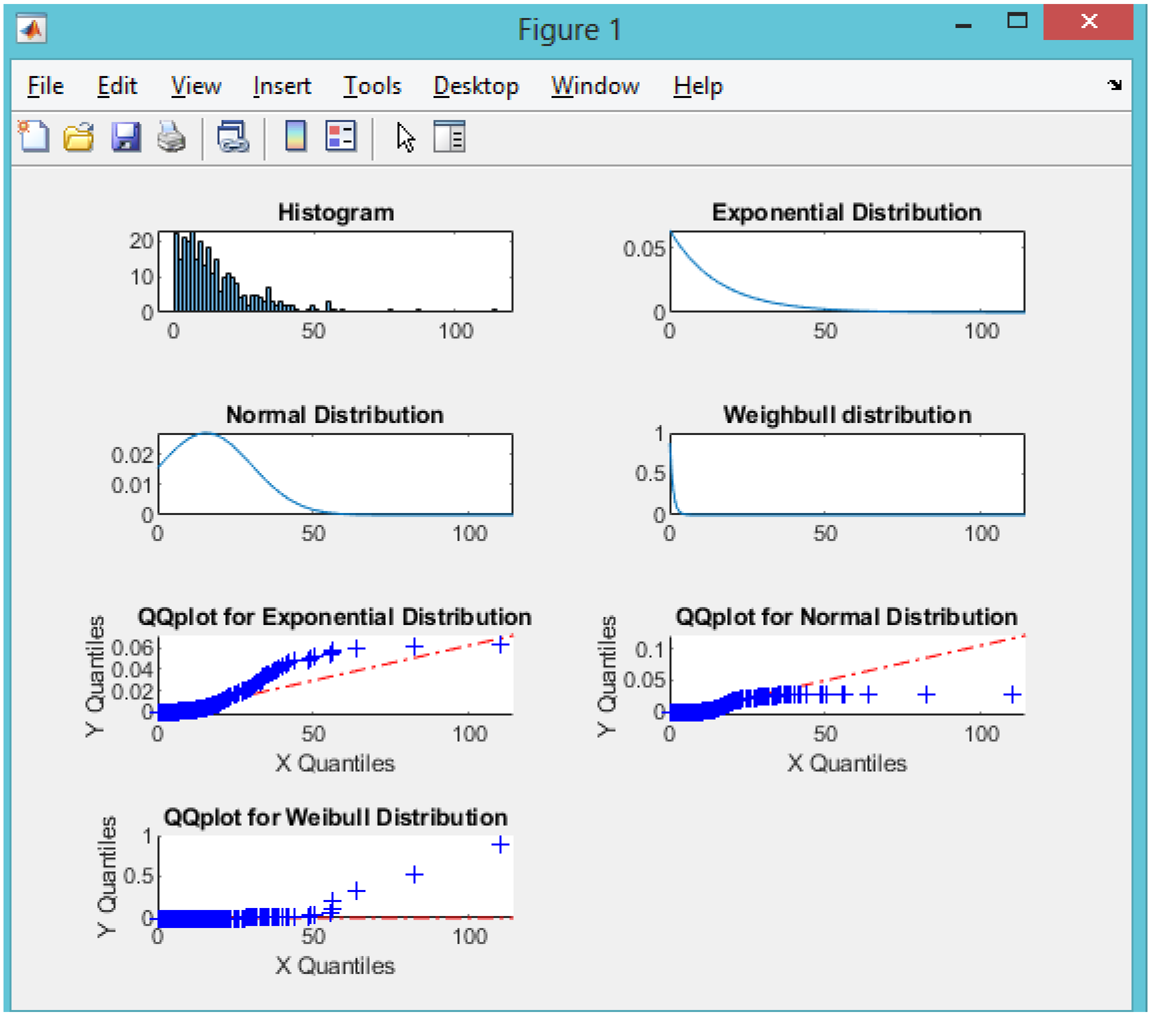
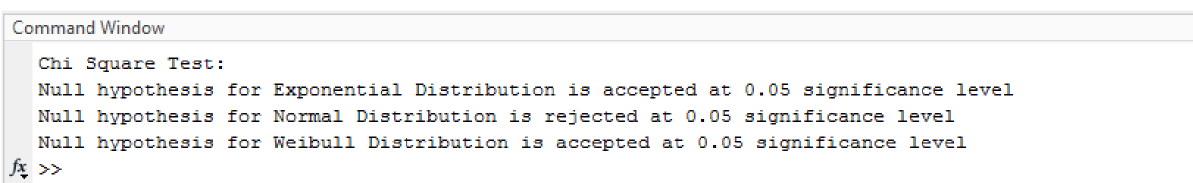


Figure 6Component 2 inspection time distribution



```
Command Window
Chi Square Test:
Null hypothesis for Exponential Distribution is accepted at 0.05 significance level
Null hypothesis for Normal Distribution is rejected at 0.05 significance level
Null hypothesis for Weibull Distribution is accepted at 0.05 significance level
>>
```

Figure 7 Null hypothesis for inspection time

Distribution of Inspection time of component 3:

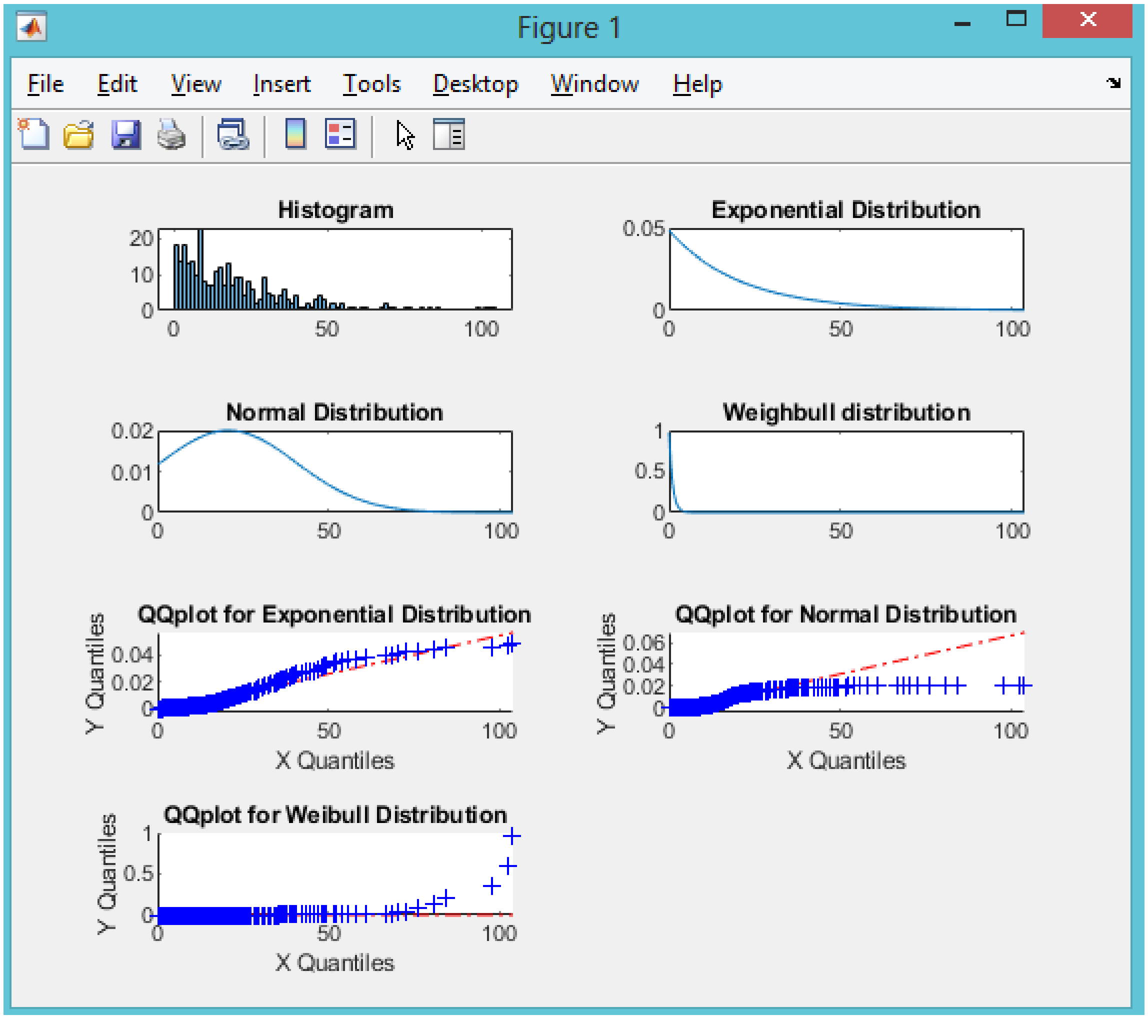


Figure 8Component 3 inspection time distribution

Command Window

```
Chi Square Test:
Null hypothesis for Exponential Distribution is accepted at 0.05 significance level
Null hypothesis for Normal Distribution is rejected at 0.05 significance level
Null hypothesis for Weibull Distribution is accepted at 0.05 significance level
>>
```

Figure 9Null hypothesis for component 3

Workstation 1 service time Distribution output:

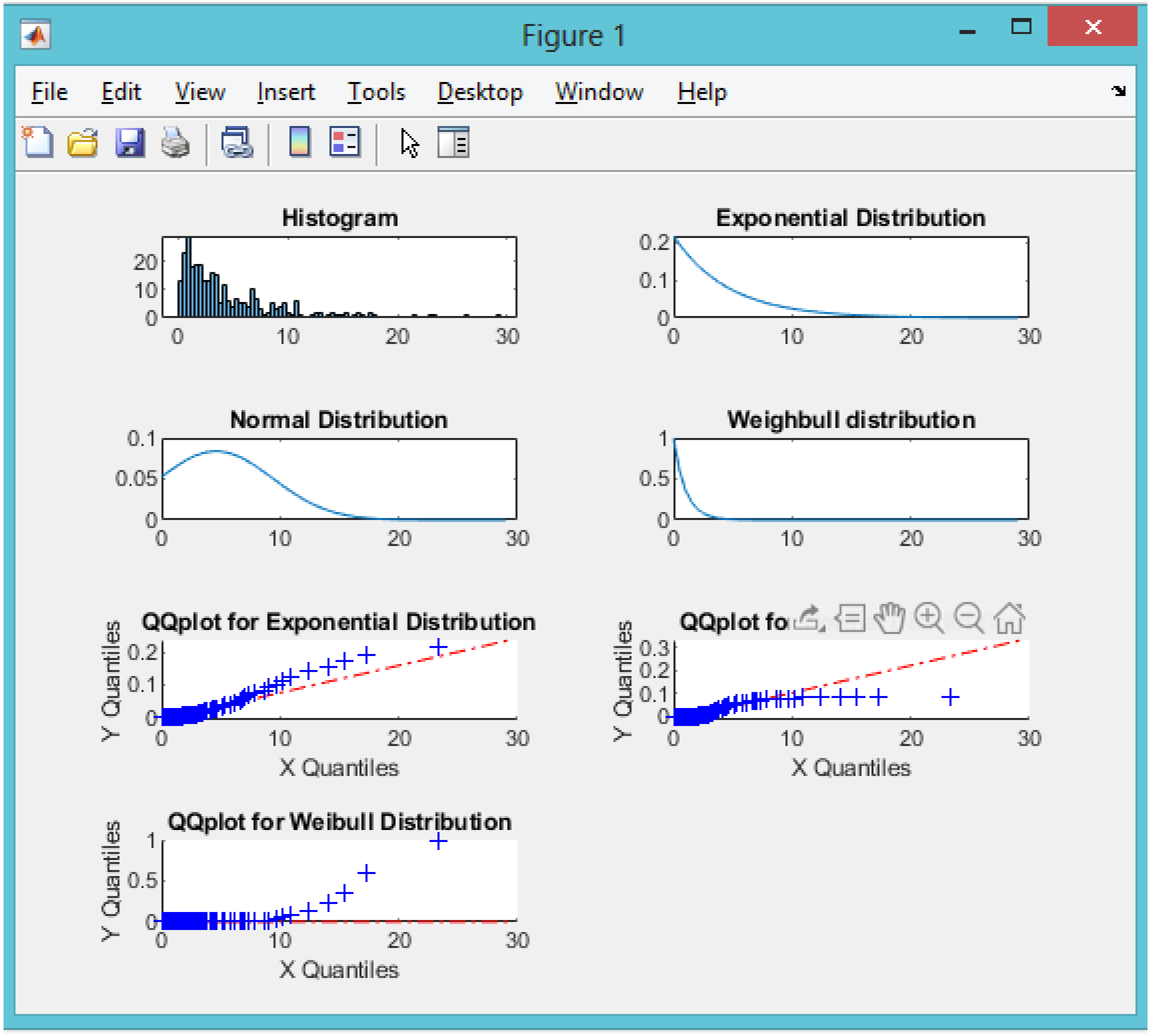


Figure 10 Workstation 1 service time distribution

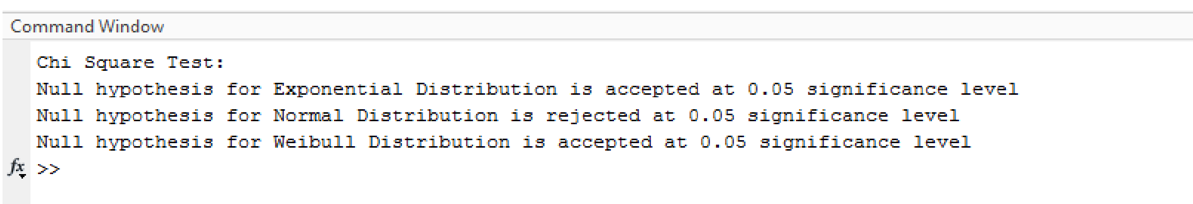


Figure 11Null hypothesis

Workstation 2 service time distribution output:

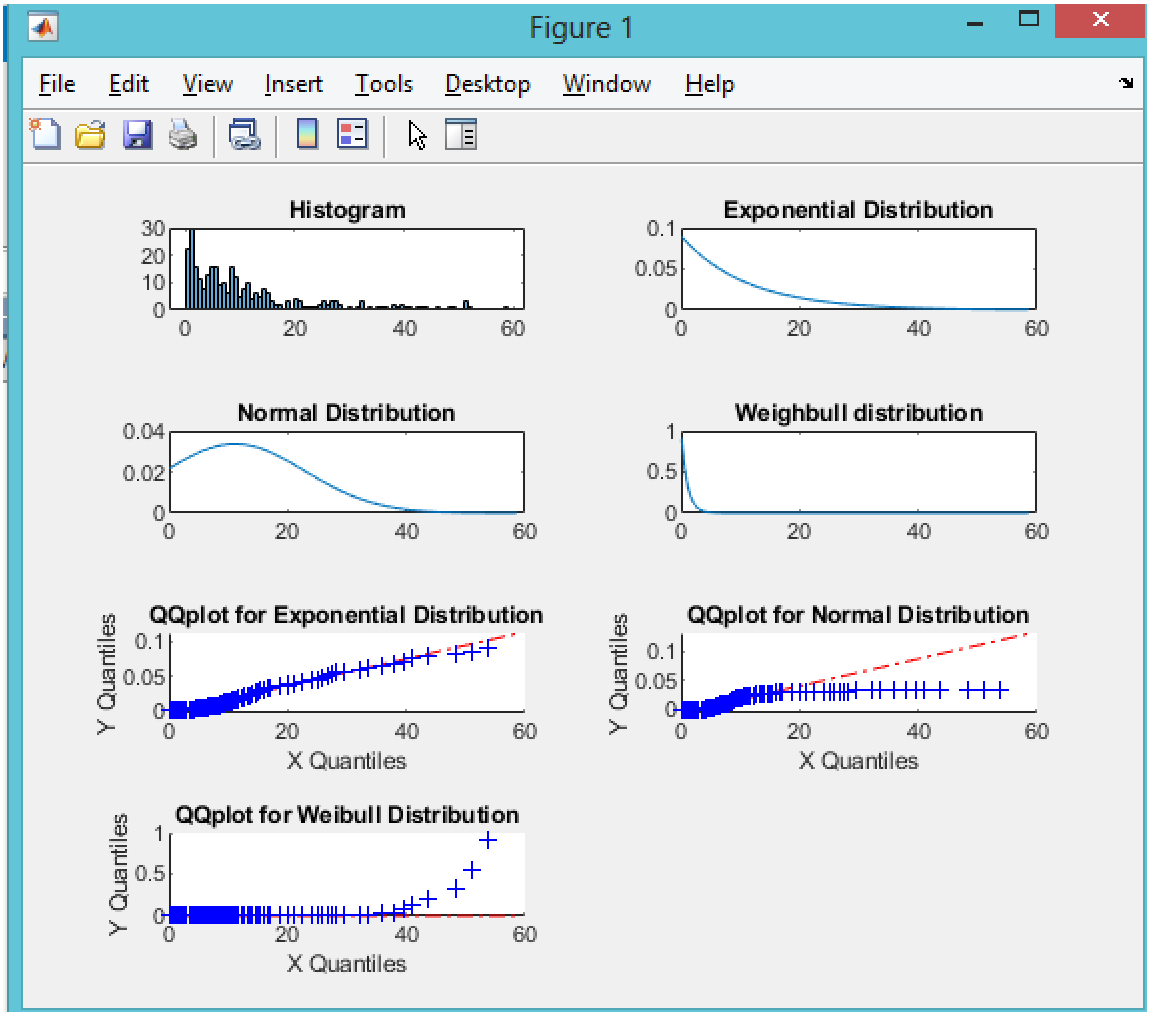


Figure 12Workstation 2service time distribution

```
Command Window
Chi Square Test:
Null hypothesis for Exponential Distribution is rejected at 0.05 significance level
Null hypothesis for Normal Distribution is rejected at 0.05 significance level
Null hypothesis for Weibull Distribution is accepted at 0.05 significance level
>>
```

Figure 13 Null Hypothesis for workstation 2 service time

Workstation 3 service time distribution output:

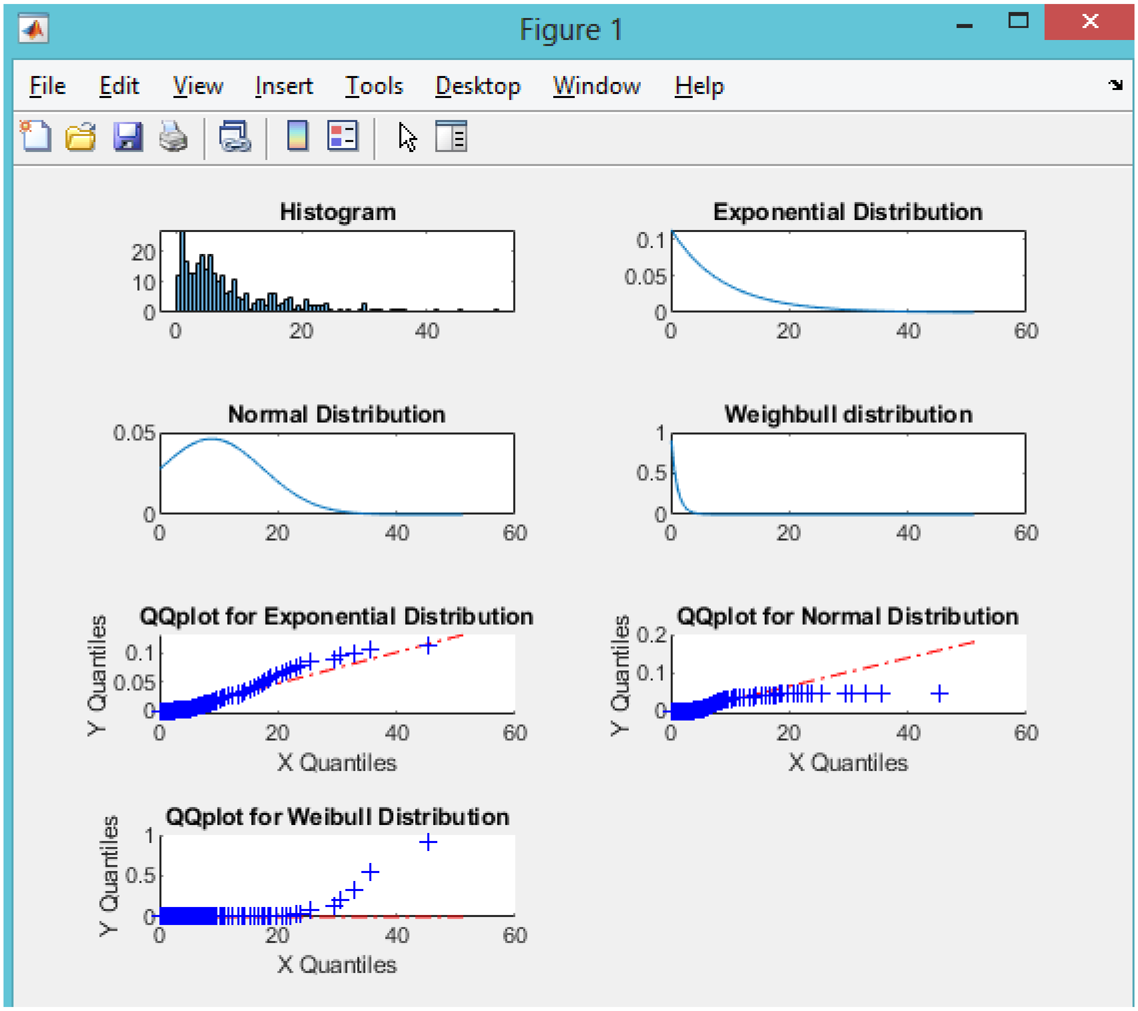


Figure 14Workstation 3 service time distribution

```
Command Window
Chi Square Test:
Null hypothesis for Exponential Distribution is accepted at 0.05 significance level
Null hypothesis for Normal Distribution is rejected at 0.05 significance level
Null hypothesis for Weibull Distribution is accepted at 0.05 significance level
fx >>
```

Figure 15 Null hypothesis for service time of workstation 3

Generating Input based on the model Identified:

To generate input based on the model identified in deliverable 1, I have used inverse transform technique as the data set is discrete. All discrete distribution can be generated using this method. Below mentioned is the code for inverse transform technique in which I have used exponential distribution and displayed the output of histogram.

Histogram:

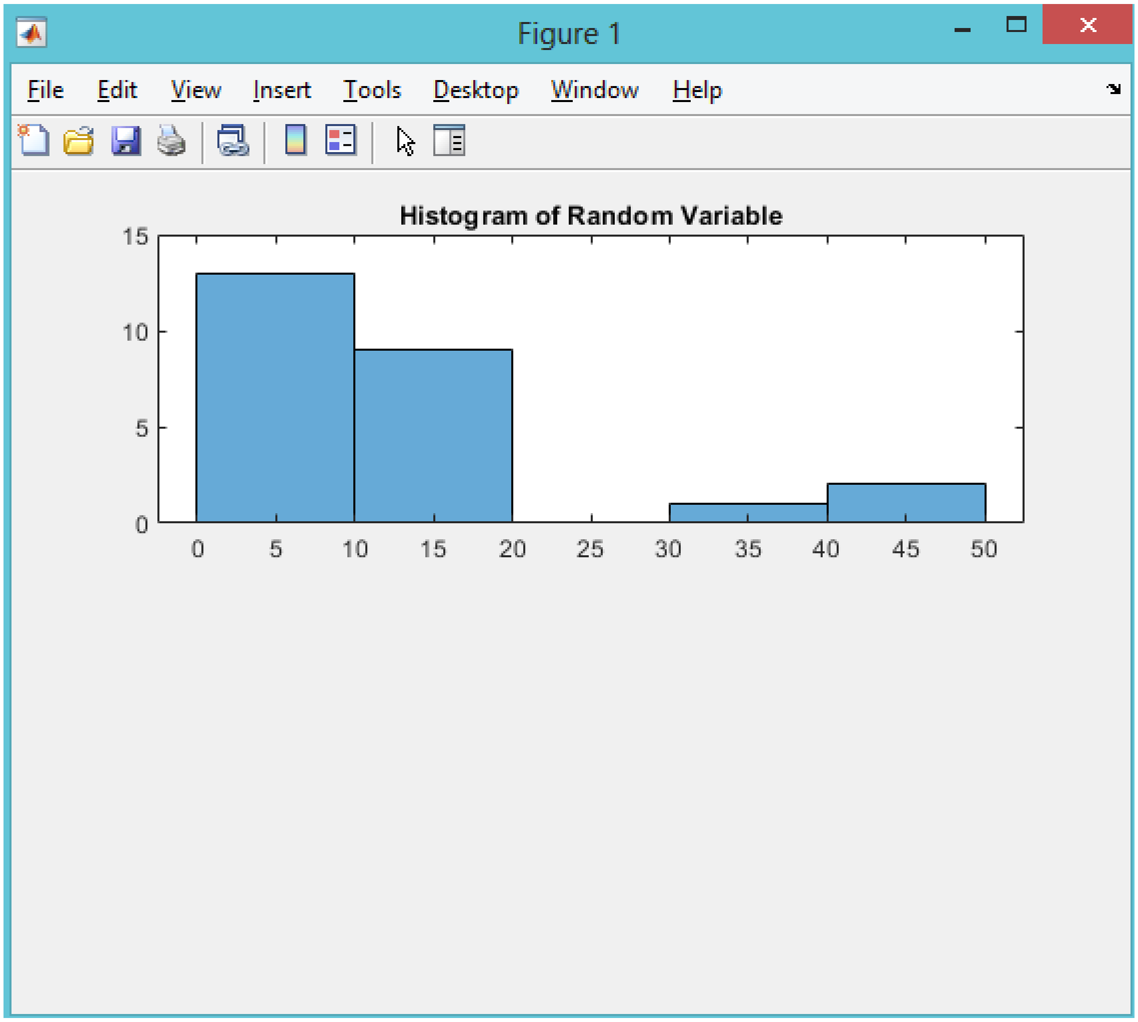


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Histogram of Random Variable
0
5
10
15
0
5
10
15
20
25
30
35
40
45
50

Deliverable 3:

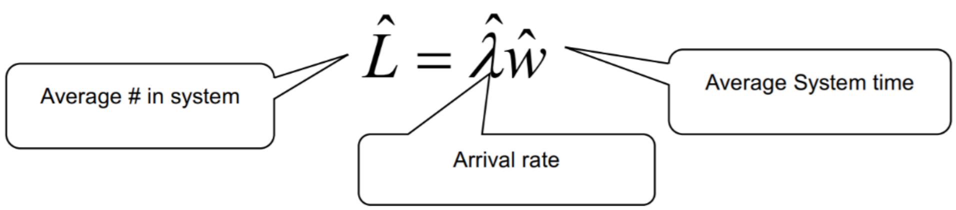
Project Description:

In this deliverable, I have to simulate the and verify the model. After doing verification, validation needs to be done. For Verification, Little’s law is mandatory for this model. I need to use Little's law for the whole system as well as for each individual queue.

For production run, I have to set a precision and then show how I decided the running length for each replication and number of replications to meet the precision requirements under 95% CI.

Little's Law works for any black box, where the black box can be a complex system, or any subunit such as a single waiting line or a single server, a waiting line plus a sever, etc.

To use Little's Law, I have to identify the black box, the input and output. Measure Lambda as the long-term input average rate, L as the average number of customers in the black box, W is the average time in the black box. Then check whether L=Lambda*W.



$$L = \lambda w \quad \text{as} \quad T \to \infty \text{ and } N \to \infty$$

Files created for the simulation:

1. **Main_Simulation.m :** This will simulate the whole manufacturing facility according to the logic given in project description.

2. **Inspector1.m :** This is a file that is used as a function in Main_simulation.m . It will give inspection time or idle time of inspector 1 and will decide the buffer logic.

3. **Inspector2.m:** This is a file that is used as a function in Main_simulation.m. It will generate inspection time or idle time of inspector 2 and will decide the buffer logic.

4. **Inspector2_c2c3:** This file is written to find Normal and exponential distribution of Inspector 2’s inspection time

5. **Inspector_c1.m:** This file is written to find Normal and exponential distribution of Inspector 1’s inspection time

Detailed level code and It's Output that are useful for this Deliverable:

1. **Little's_Law.m:** This file is used to derive Little's law for individual queue (for inspection time of inspector 1, inspection time of inspector 2, throughput for every workstation, blocking probability of every inspector and workstation).
   It is important to notice here that this file will give answer to what are the outputs , throughputs and blocking probability of inspectors in this simulation.

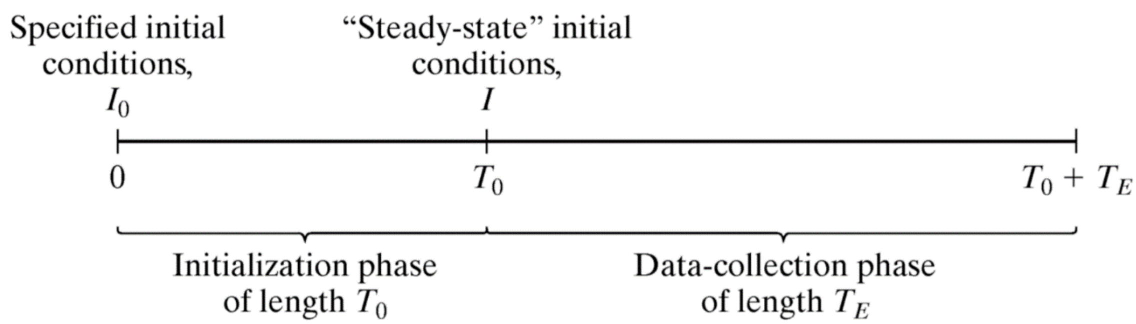
Output:

```
Editor - Littles_Law.m                Variables - steady_state
Main_Simulation.m   Littles_Law.m   steady_state_analysis.m
274
Command Window
New to MATLAB? See resources for Getting Started.
  Blocking Probability of Workstation 1 = 5.344828e-01

  Products from Workstation 2 = 26
  Service time of Workstation 2 = 250
  Idle time of Workstation 2 = 3288

  Blocking Probability of Workstation 2 = 9.296213e-01

  Throughput from Workstation 2 = 7.066139e-03
  Products = 25
  Products from Workstation 3 = 0
  Service time of Workstation 3 = 0
  Idle time of Workstation 3 = 3538

  Throughput from Workstation 3 = 0
  Products = 0
  Blocking Probability of Workstation 3 = 1.000283e+00

  Run time of Simulation = 3538

  Average Number of products for the black box system (L_Total) = 8.479367e-02

  Average Number of Items  for:
  Inspector 1 = 1
  Inspector 2 = 1.670435e-01
  Workstation 1 = 4.657999e-01
  Workstation 2 = 7.066139e-02
  Workstation 3 = 0
fx >>
```

Figure 16 Little's Law for Individual and Black box

Here it is observed that Little's law defines the L(average number of products in black box system) as 0.0847. And for Individual items, the values are correct as it can be seen that inspector 1 has high number of average time of inspection than inspector 2 . And workstation 3 will have 0 processing time as no products are created by workstation 3.

2. **steady_state_analysis.m:** This file is used to give steady state analysis in which T0 and TE time parameters are measured. It is observed that here, for this simulation T0= 350 second, data collection phase starts. Here initialization bias is from 0 to T0. And TE extends to 3539 for this simulation which takes some time and gives good result.



Steady State Output:

```
Command Window
New to MATLAB? See resources for Getting Started.
Steady State starts after T0=350 and extends upto TE=3539
 Values For Steady State:
Inspection time of Inspector 1 = 3179
Idle time of Inspector 1 = 0
Inspection time of Inspector 2 = 375
Idle time of Inspector 2 = 2781
Service time of workstation 1 = 1491
Service time of workstation 2 = 227
Service time of workstation 3 = 0
Idle time of workstation 1 = 1688
Idle time of workstation 2 = 2952
Idle time of workstation 3 = 3179
fx >>
```

Figure 17Steady State Output

Here I ran the code whole simulation time which includes transient and steady state time and calculated the value for above mentioned parameters.

Production Runs and Analysis:

For this part I have created multiple files to find the best confidence interval and worst confidence intervals and took input from the defined data in the project and randomized it to normal distribution . So each file will have normal distribution and based on that I have defined the output of worst case and best case scenario for particular CI interval. More Replications are used to minimize the errors occurred in the simulation.

Random generation of input is also based on the normal distribution which can be seen in the first part of the code.

$$\bar{Y}_{..} \pm t_{\alpha/2,R-1} \frac{S}{\sqrt{R}}$$

This is the formula I used to define the Confidence interval. This is achieved by using paramci function for normal distribution which will give upper and lower bound for CI.

To generate random numbers, I took the data given in this project and randomized it in every code to check confidence interval.

OutPut for CI for Inspection time of Component 1:

```
Command Window
New to MATLAB? See resources for Getting Started.
  CI (95 percentage)for 7 replications :  [8.851725e+00,8.892650e+00]
  Best Case Scenario for these replications = 1.465260e+00
  Worst Case Scenario for these replications= 1.506185e+00
fx >>
```

Figure 18 CI for inspection time of C1

OutPut for CI for Inspection time of Component 2:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 20 replications : [1.857756e+01,1.868255e+01]
Best Case Scenario for these replications = 1.950208e+00
Worst Case Scenario for these replications= 2.055192e+00
fx >>
```

Figure 19 CI for inspection time of C2

OutPut for CI_for_Inspection_time_C3:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 15 replications : [1.367481e+01,1.399519e+01]
Best Case Scenario for these replications = 1.541712e+00
Worst Case Scenario for these replications= 1.862090e+00
fx >>
```

Figure 20 CI for inspection time of C3

OutPut for CI_for_Workstation1:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 12 replications : [1.087939e+01,9.601248e+00]
Best Case Scenario for these replications = 1.491358e+00
Worst Case Scenario for these replications= 2.132208e-01
fx >>
```

Figure 21 CI for workstation 1 processing time

OutPut for CI_for_Workstation2:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 6 replications :  [4.449650e+00,4.137308e+00]
Best Case Scenario for these replications = 4.671088e-01
Worst Case Scenario for these replications= 1.547669e-01
fx >>
```

Figure 22 CI for workstation 2 processing time

OutPut for CI_for_Workstation3:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 9 replications :  [4.404015e+00,4.063977e+00]
Best Case Scenario for these replications = 5.404392e-01
Worst Case Scenario for these replications= 2.004018e-01
fx >>
```

Figure 23 CI for workstation 3 processing time

Deliverable 4:

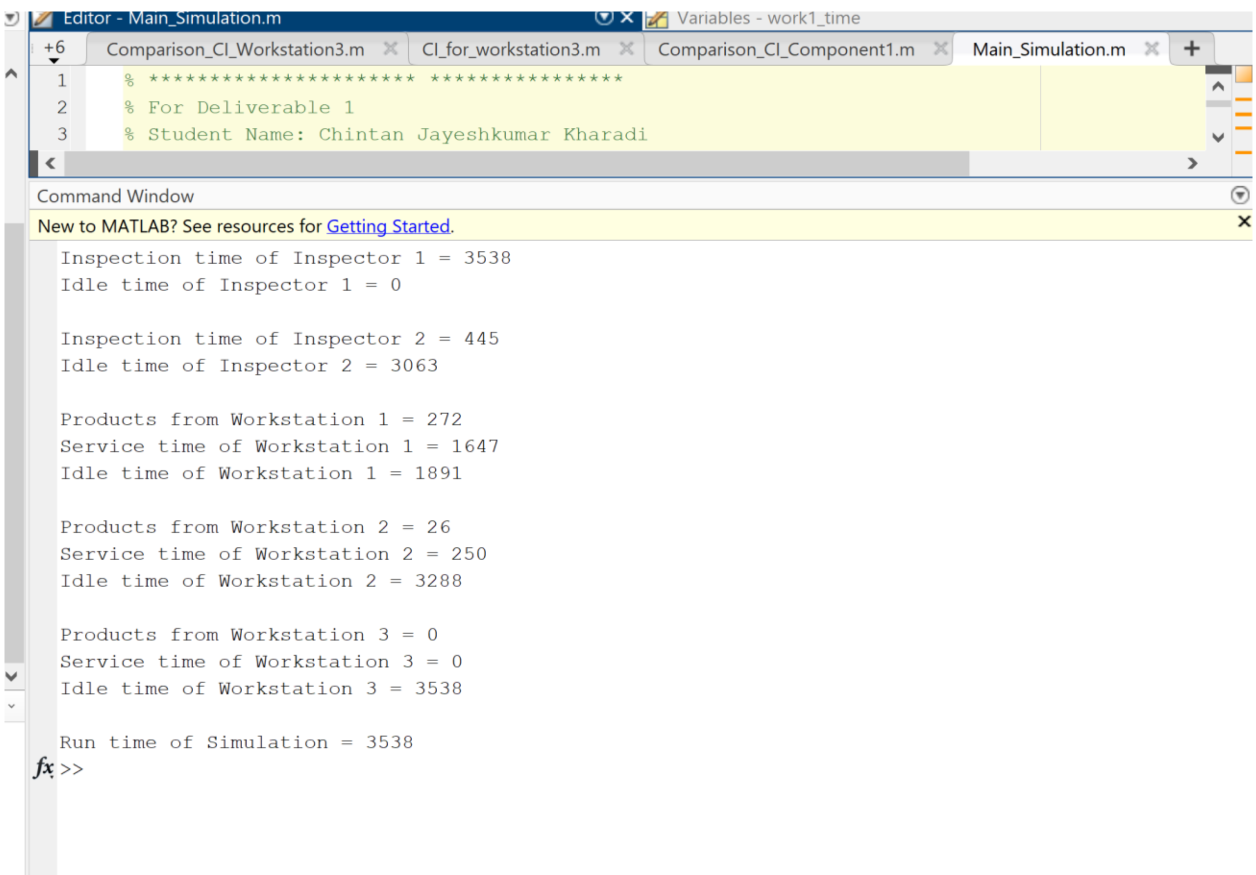
Project Description:

In this deliverable, proposal of alternative design is essential. This alternative model needs to verified and validated using the same process that has been done in deliverable 3. In addition to it, comparison of alternative and original model must be done using common random numbers (CRN method) with synchronization.

Mention the conclusion of new alternative design whether it is better, worse or the same.

Original Model Simulation:

The Original model produces numerous product 1 but it is clearly visible that product 2 is produced less and product 3 is not created in this model. The outcome of this simulation is displayed as below:



Products from workstation 1: 272

Products from workstation 2: 26

Products from workstation 3: 0

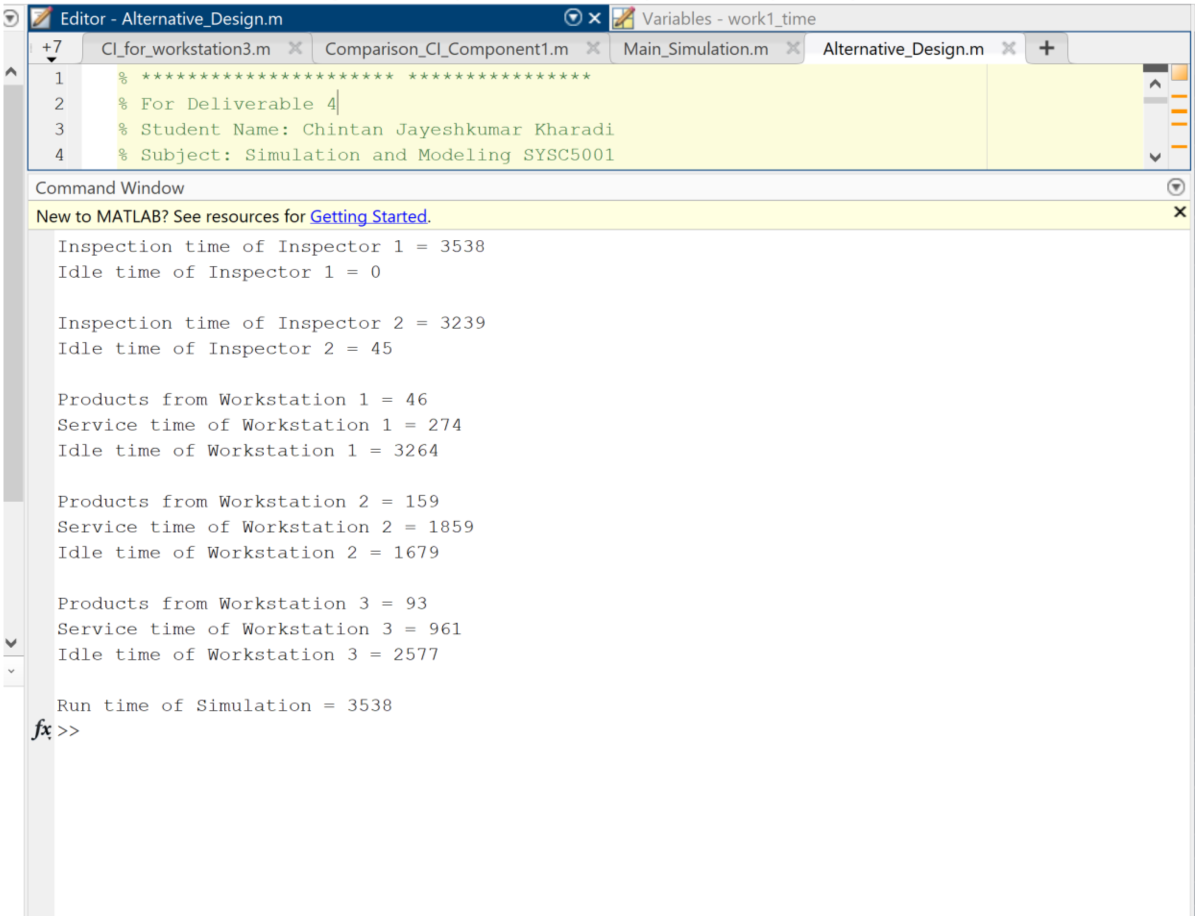
This result is because of the original model design. Component 1 is always used in this 3 products and if it is not there, no products can be produced. In this original model, priority to select component 1 from 3 buffers is given to the buffer 1 of workstation 1 given as highest, then moderate priority to workstation 2 and lowest priority to workstation 3. So, this simulation will always converge to produce higher number of product 1 and less number of product 2. Workstation 3 will not get any chance to produce product 3 as component 1 is not available as it is consumed by buffers of workstation 1 and 2.

So, I have created an alternative design to produce more number of product 2 and product 3 than product 1.

Alternative Design:

Previous model was more focused on giving priority to first workstation's buffer, while in this design I have given highest priority to buffer of workstation 3 for component 1, then to buffer of workstation 2 for component 1 and least priority to buffer of workstation 1 for component 1. This design is assumed to produce more number of product 3 compared to the original model.

The alternative design code is in Alternative_Design.m and the result of this simulation is shown as below:



Here,

Products form Workstation 1=46

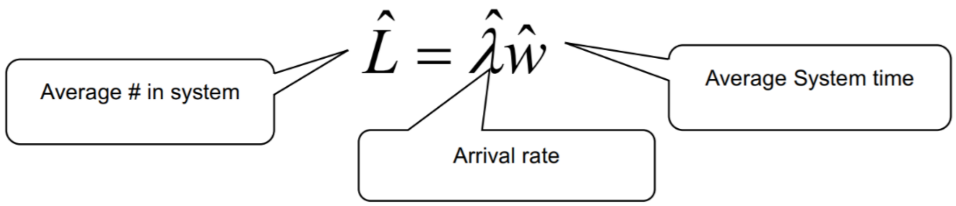
Products form Workstation 2=159

Products form Workstation 3=93

This clearly shows that this alternative design is better than previous model as it produces all three products. Previously product 3 count was 0 compared to this model in which 93 products are produced. So, if we give highest priority on buffer of component 1 of workstation 3, all three workstation has moderate priority to generate all three products.

Verification using Little's Law for Alternative Design:

As described in deliverable 3, To use Little's Law, I have to identify the black box, the input and output. Measure Lambda as the long-term input average rate, L as the average number of customers in the black box, W is the average time in the black box. Then check whether L=Lambda*W.



$$L = \lambda w \quad \text{as} \quad T \to \infty \text{ and } N \to \infty$$

The output for original model is given as below:

```
Editor - Littles_Law.m                Variables - steady_state
Main_Simulation.m   Littles_Law.m   steady_state_analysis.m
Command Window
New to MATLAB? See resources for Getting Started.
Blocking Probability of Workstation 1 = 5.344828e-01

Products from Workstation 2 = 26
Service time of Workstation 2 = 250
Idle time of Workstation 2 = 3288

Blocking Probability of Workstation 2 = 9.296213e-01

Throughput from Workstation 2 = 7.066139e-03
Products = 25
Products from Workstation 3 = 0
Service time of Workstation 3 = 0
Idle time of Workstation 3 = 3538

Throughput from Workstation 3 = 0
Products = 0
Blocking Probability of Workstation 3 = 1.000283e+00

Run time of Simulation = 3538

Average Number of products for the black box system (L_Total) = 8.479367e-02

Average Number of Items  for:
Inspector 1 = 1
Inspector 2 = 1.670435e-01
Workstation 1 = 4.657999e-01
Workstation 2 = 7.066139e-02
Workstation 3 = 0
fx >>
```

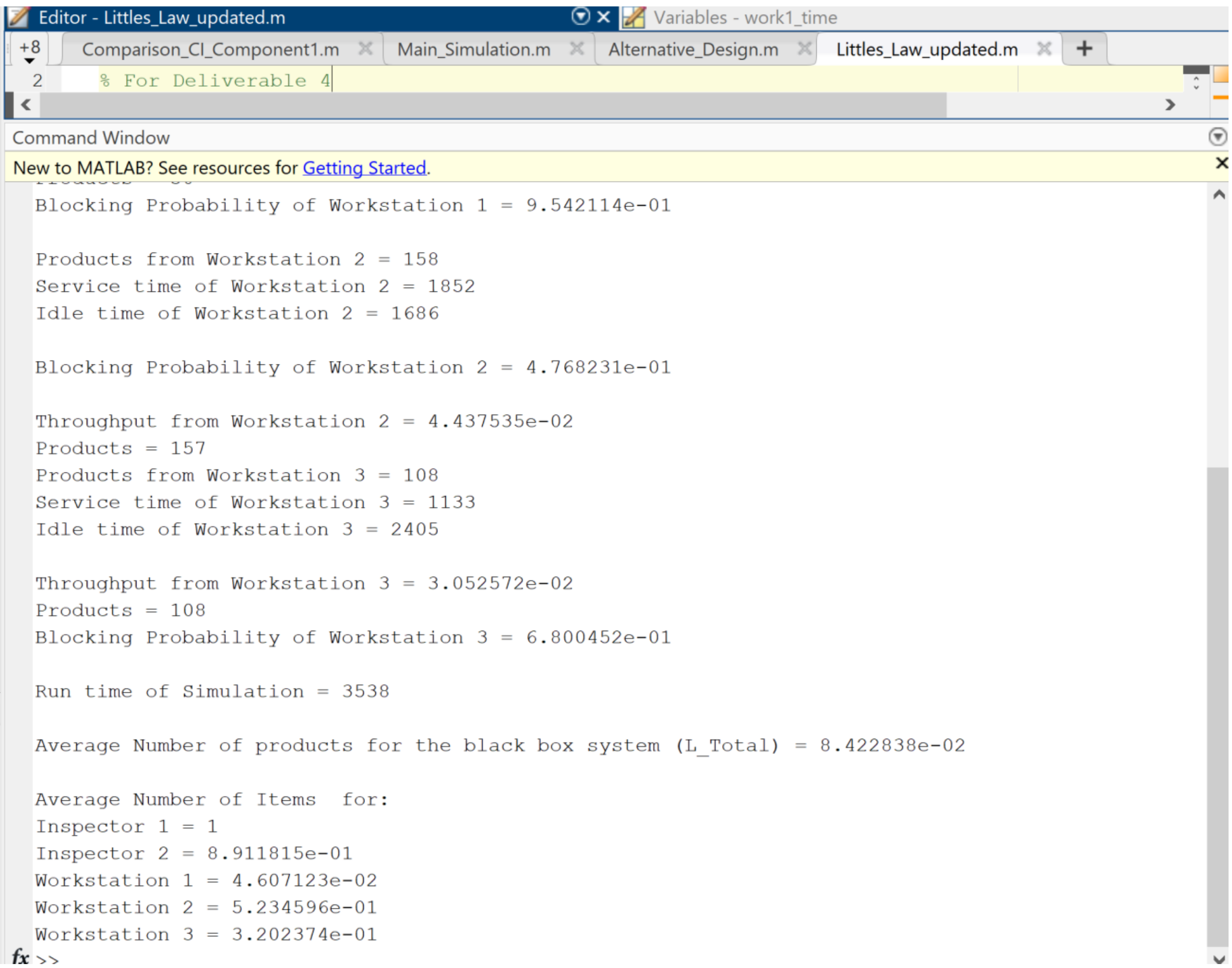
Here,

Average number of products coming from Workstation 1=0.465799

Average number of products coming from Workstation 2=0.07066

Average number of products coming from Workstation 3=0

For, Alternative model, I have created new file, Littles_Law_updated.m. The result of this is given as below:



From this, I can identify now that average number of products that remain in black box system is similar to the original model. So, this alternative design is not impactful to create new products but it does shift focus from workstation 1 to workstation 3 to create product 3.

Average number of products from workstation 1=0.046071

Average number of products from workstation 2=0.52345

Average number of products from workstation 3=0.3202

When I apply little's law for individual, then main difference is visible for inspector 2 , workstation 2 and workstation 3 and it is increased which is a good sign for this model. While, products from workstation 1 has decreased because of this new design.

Production Runs and Analysis:

For this part I have created multiple files to find the best confidence interval and worst confidence intervals same as done in deliverable 3and took input from the defined data given in the project and randomized it to normal distribution. So each file will have normal distribution and based on that I have defined the output of worst case and best case scenario for particular CI interval. More Replications are used to minimize the errors occurred in the simulation.

Random generation of input is also based on the normal distribution which can be seen in the first part of the code.

To generate random numbers, I took the data given in this project (for e.g. ws1.dat,servinsp1.dat)and randomized it in every code to check confidence interval. And normal distribution is used in randomizing the data.

$$\bar{Y}_{..} \pm t_{\alpha/2,R-1} \frac{S}{\sqrt{R}}$$

This is the formula I used to define the Confidence interval. This is achieved by using paramci function for normal distribution which will give upper and lower bound for CI.

CI for component 1:

```
Command Window
New to MATLAB? See resources for Getting Started.
  CI (95 percentage)for 15 replications : [1.367481e+01,1.399519e+01]
  Best Case Scenario for these replications = 1.541712e+00
  Worst Case Scenario for these replications= 1.862090e+00
fx >>
```

CI for component 2:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 7 replications :  [1.183278e+01,1.183278e+01]
Best Case Scenario for these replications = 1.474866e+00
Worst Case Scenario for these replications= 1.474866e+00
fx >>
```

CI for component 3:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 6 replications :  [4.449650e+00,4.137308e+00]
Best Case Scenario for these replications = 4.671088e-01
Worst Case Scenario for these replications= 1.547669e-01
fx >>
```

CI for workstation 1:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 20 replications : [1.857756e+01,1.868255e+01]
Best Case Scenario for these replications = 1.950208e+00
Worst Case Scenario for these replications= 2.055192e+00
fx >>
```

CI for workstation 2:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 12 replications : [1.087939e+01,9.601248e+00]
Best Case Scenario for these replications = 1.491358e+00
Worst Case Scenario for these replications= 2.132208e-01
fx >>
```

CI for workstation 3:

```
Command Window
New to MATLAB? See resources for Getting Started.
CI (95 percentage)for 9 replications :  [4.404015e+00,4.063977e+00]
Best Case Scenario for these replications = 5.404392e-01
Worst Case Scenario for these replications= 2.004018e-01
fx >>
```

Comparison of Original and Alternative Model:

Comparison of both model can be done using below mentioned method:

This method will tell us how good or bad alternative model is. I am going to use common random number (CRN) on both the models at once to check whether the alternative model is better or not.

This comparison is done on workstations as the workstation are the one which creates the products.

Steps to compare these systems:

The estimator based on CRN is more precise, leading to a shorter confidence interval for the difference: $D_r = \bar{Y}_{r1} - \bar{Y}_{r2}$

Sample average: $\bar{D} = \frac{1}{R}\sum_{r=1}^{R} D_r$

Sample variance of the differences:

$$S_D^2 = \frac{1}{R-1}\sum_{r=1}^{R}(D_r - \bar{D})^2 = \frac{1}{R-1}\left[\sum_{r=1}^{R} D_r^2 - R\bar{D}^2\right]$$

Standard error:

$$s.e.(\bar{D}) = s.e.(\bar{Y}_{.1} - \bar{Y}_{.2}) = \frac{S_D}{\sqrt{R}}$$

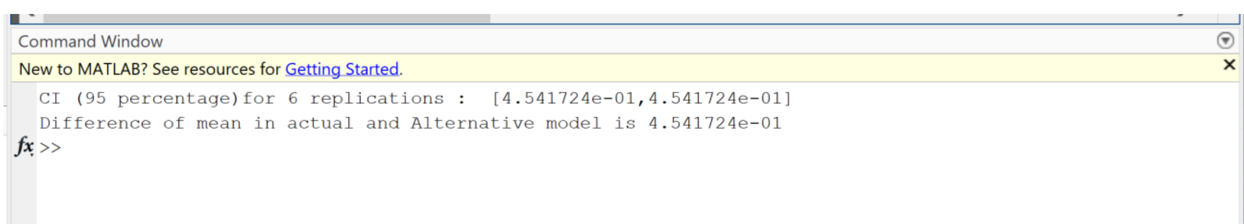
Degree of freedom:

$$v = R - 1$$

Based on CI the result can be manipulated using below mentioned method:

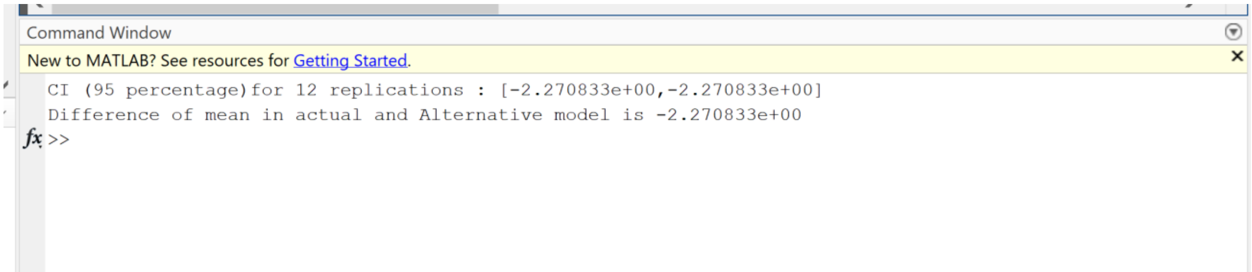
If c.i. is totally to the left of *0*, strong evidence for the hypothesis that $\theta_1 - \theta_2 < 0$ ($\theta_1 < \theta_2$).
If c.i. is totally to the right of *0*, strong evidence for the hypothesis that $\theta_1 - \theta_2 > 0$ ($\theta_1 > \theta_2$).
If c.i. is totally contains *0*, no strong statistical evidence that one system is better than the other

Comparison for Workstation 1:


Command Window
New to MATLAB? See resources for Getting Started.
```
CI (95 percentage)for 6 replications :  [4.541724e-01,4.541724e-01]
Difference of mean in actual and Alternative model is 4.541724e-01
fx >>
```

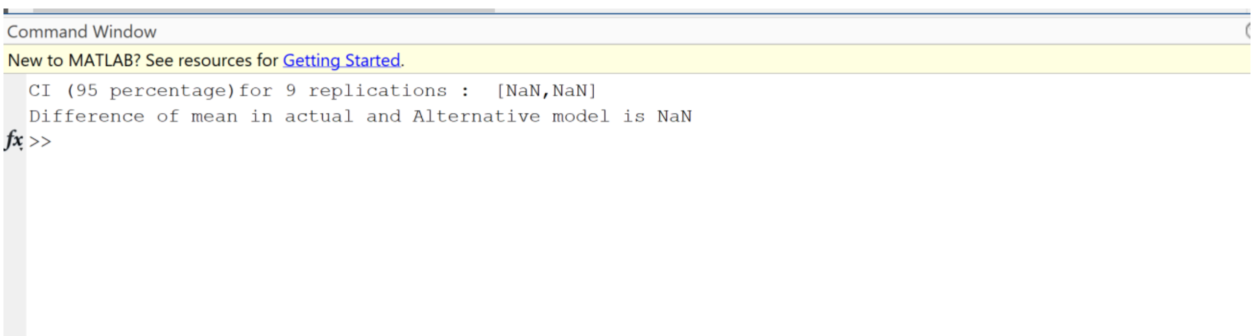
Here positive difference shows that actual system is faster than the alternative model by 45.4 times which is very high as product 1 was created many times in the original model than the alternative model.

Comparison for Workstation 2:



Here the difference is negative which shows that the alternative model is faster than the actual model by 227 seconds as it produces more number of products 2 in alternative model than in the actual model.

Comparison for Workstation 3:



Here the difference is NaN because actual model was not giving any products and because of that I can say that alternative model is infinite times faster than the actual model as it was not making product 3.

Conclusion:

Below mention results will tell which system is better:

Actual Model:

These data is approximation as it will change little bit everytime based on the inspection time.

Product 1 count= 272

Product 2 count = 26

Product 3 count=0

Alternative Model:

Product 1 count= 46

Product 2 count = 159

Product 3 count=93

It is clearly visible that the alternative model is far better than the actual model as it was not creating any product 3 while in alternative model the count of product 3 is 93. The policy used to create product 3 in alternative model increased the productivity of workstation 3 and blocking probability of inspector 2 is reduced as workstation 2 and 3 are working properly.

In above mentioned comparison of CI, it is obvious that the ratio of producing product 2 and 3 has increased significantly while the product 1 is decrease due to this routing policy. So, this alternative model is far better as all 3 products are created and workstations are working and workstation's idle time and inspector's blocking probability is also reduced.

References:

(1) J. Banks, J. C. Carson II, B. L. Nelson, and D. M. Nicol, Discrete-Event System Simulation, 5/E, ISBN- 10:0136062121, ISBN-13: 9780136062127, Pearson, 2010