VALIDATING AND VERIFYING SIMULATION OF QUEUING SYSTEMS(A/B/C)

Smriti Saryu Puppala   
Student, Master’s Programme in Telecommunication Systems  
*Blekinge Institute of Technology*Karlskrona, Sweden  
[smpu22@student.bth.se](file:///C:\Users\Smriti%20Saryu%20Puppala\Desktop\BTH\LP-4\NETWORK%20&%20SERVICE%20OPERATIONS\Project\smpu22@student.bth.se)

Tanishq Reddy Gouni   
Student, Master’s Programme in Telecommunication Systems  
*Blekinge Institute of Technology*Karlskrona, Sweden  
[tago22@student.bth.se](file:///C:\Users\Smriti%20Saryu%20Puppala\Desktop\BTH\LP-4\NETWORK%20&%20SERVICE%20OPERATIONS\Project\tago22@student.bth.se)

# Introduction

This report shows what our group validated while exploring different queuing systems using the provided software. We used the mm1.exe simulator to answer specific questions about these systems. We followed some basic rules for the simulation but got to choose other details ourselves. The report explains how we concluded the hypothesis for the simulations, and it includes tables and graphs with relevant information. Our main goal is verification and validation of simulation models.

# Project Description -Part A

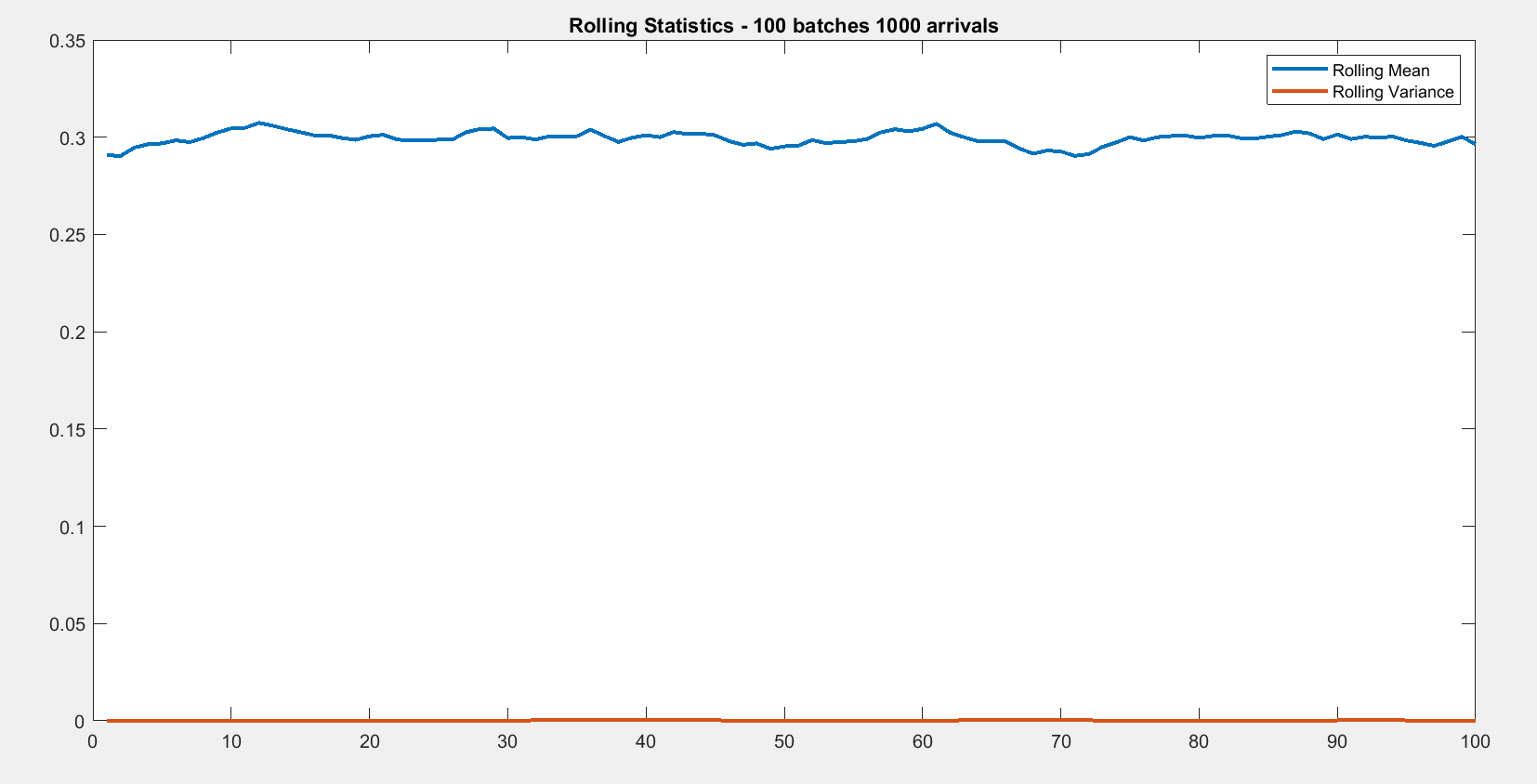
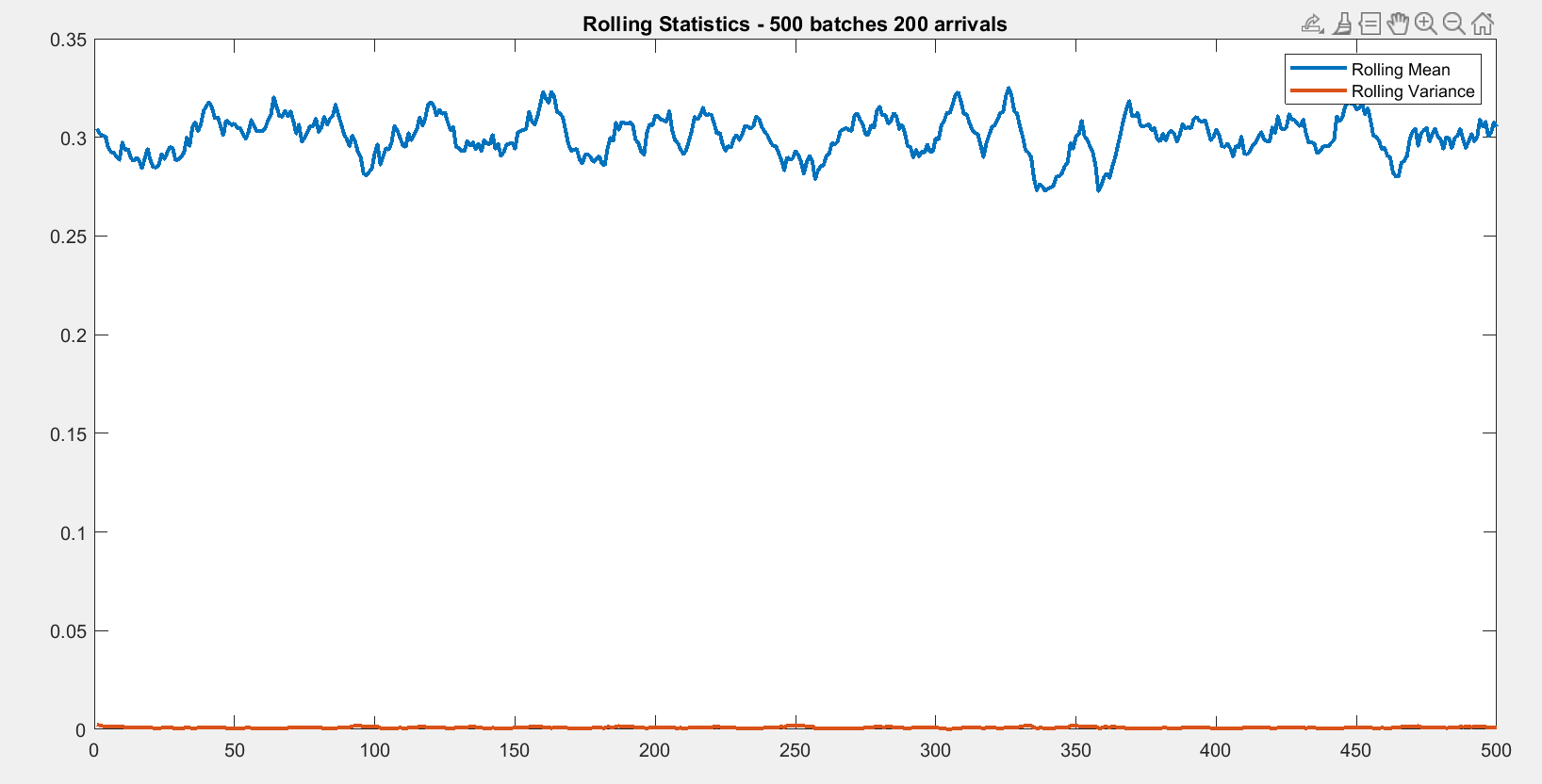
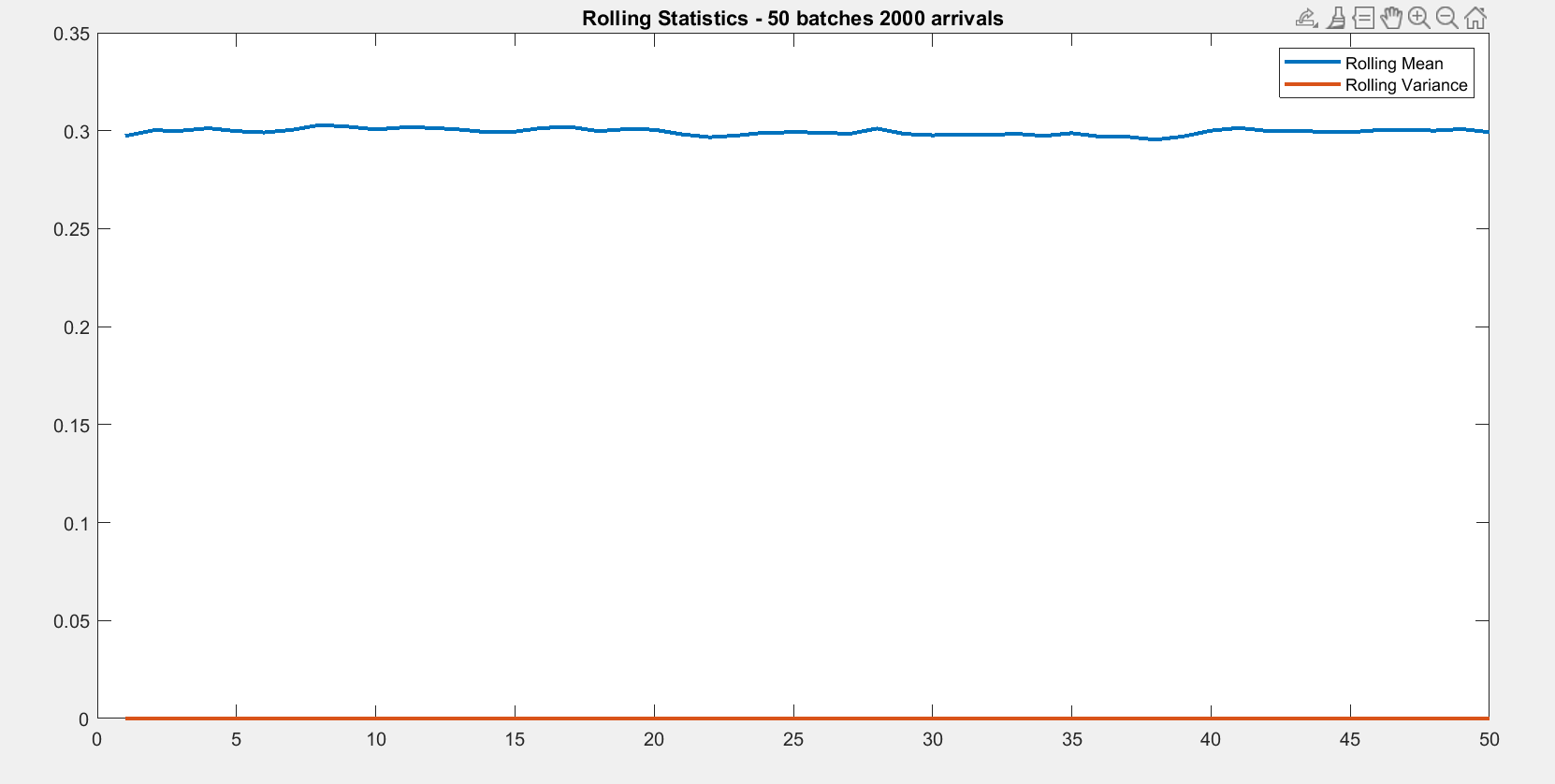
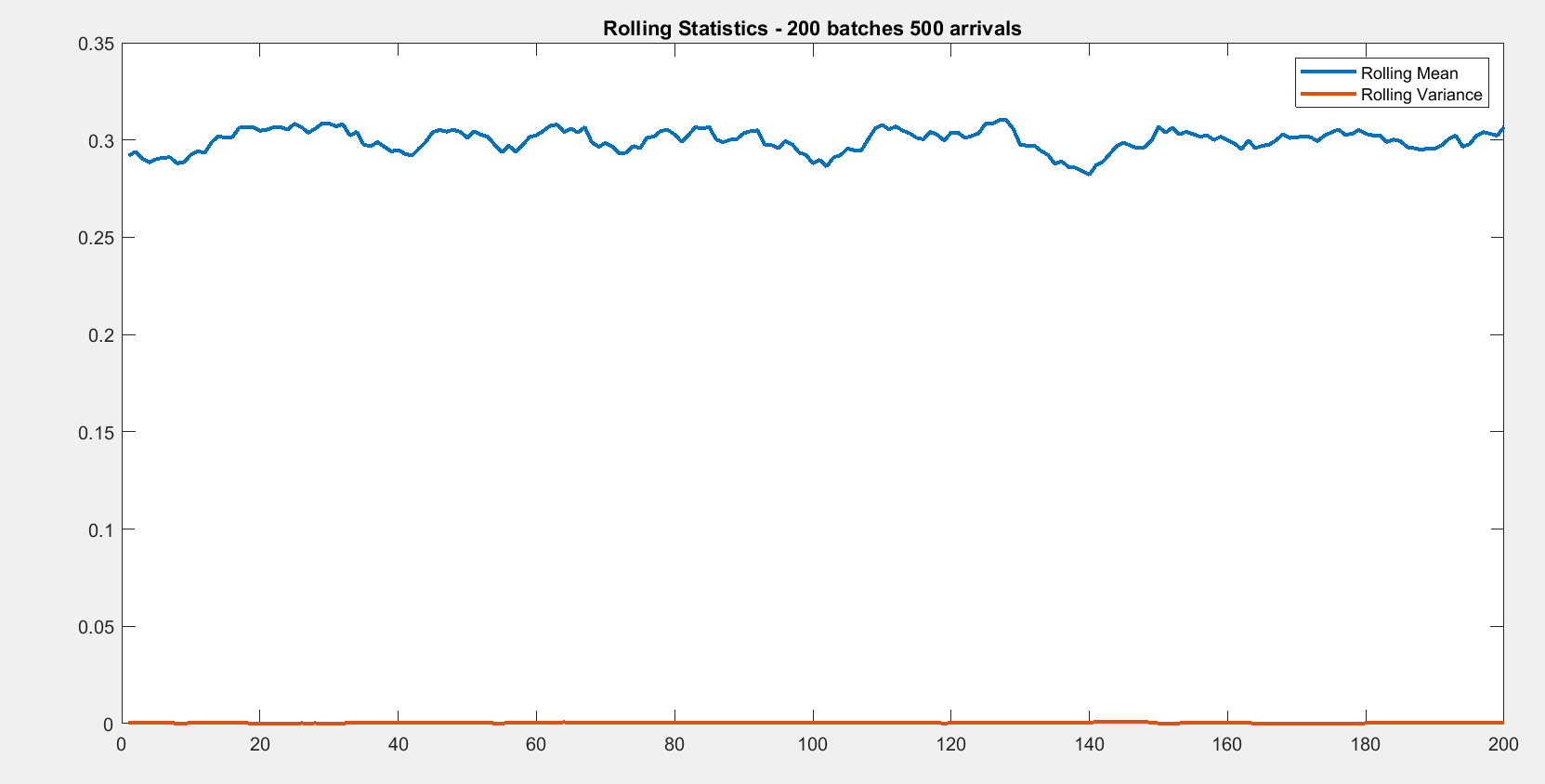
The task involves simulating different queuing systems labeled as A/B/C using the mm1.exe simulator for Task-2. The mm1.exe is run using Microsoft Visual Studio 2017,after configuring the necessary file locations ,

We rebuild the solution and run the local debugger to open debugger window to give specifications of the queuing system to be simulated, The specifications of the systems are changed according to the data required to answer the mentioned questions. We have given our hypothesis in bold and italics, while validation in bold and blue in the document.

# Analysis of the Systems

## **EXPERIMENT -1**

## **Warm Up** For considering Warm up, Covariance Stationary method was considered, rolling statistics over different windows over time series has been checked to validate the stability in the system so as to determine the system time and omit the setup time. In a covariance stationary time series, the rolling mean should remain relatively constant over different windows. Significant upward or downward trends in the rolling mean and variance may indicate non-stationarity.



*Fig 1,2,3,4 – Rolling Statistics for different number of batches and length of batches*

From the above observations, we do not wish to see an already warm system nor have the conditions when transient values are considered. A safe option is to go for 200 batches, 500 arrivals, even after 1st batch is removed. And simulation has been stopped when variance has been extremely small.

|  |  |
| --- | --- |
| Fig. Cannot consider until steady  state reached | Fig. Already Warmed up system |

*Fig 5- Warm Up*

## **EXPERIMENT-2(Q1 &Q2)**

**IMPACT ON AUTOCORRELATION**

The parameters considered for evaluating the impact on Auto Correlation are

Seed: 1509

System: M/M/1  
Inter Arrival Time: 100

Avg Service Time: 30  
Queue Length: 10

Warmup: 1   
We have set the sample length to be 100,000.  
Keeping the sample length constant we have changed the batch length and number of arrivals accordingly as

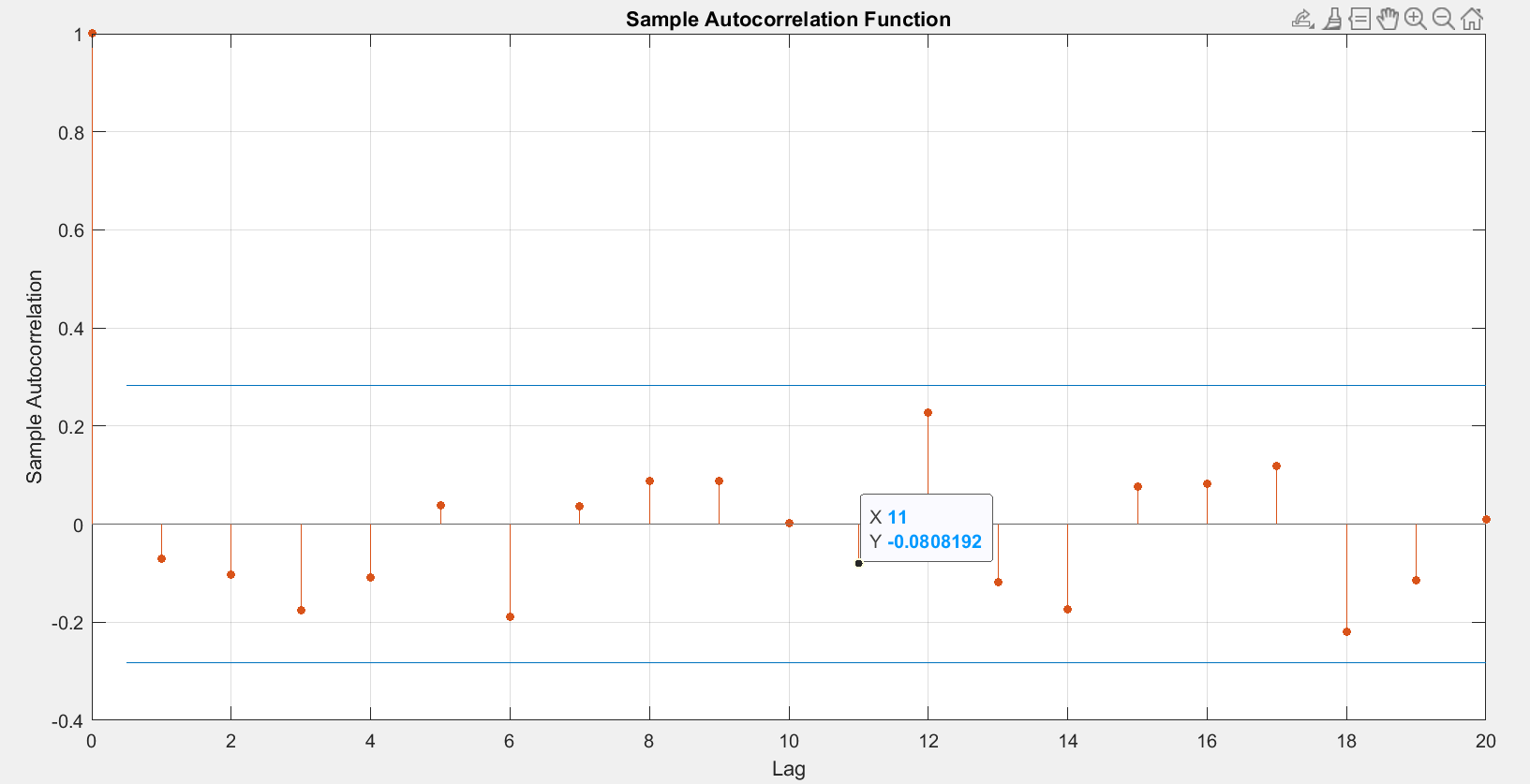
Number of batches:50 Length of batches:2000

Number of batches:100 Length of batches:1000

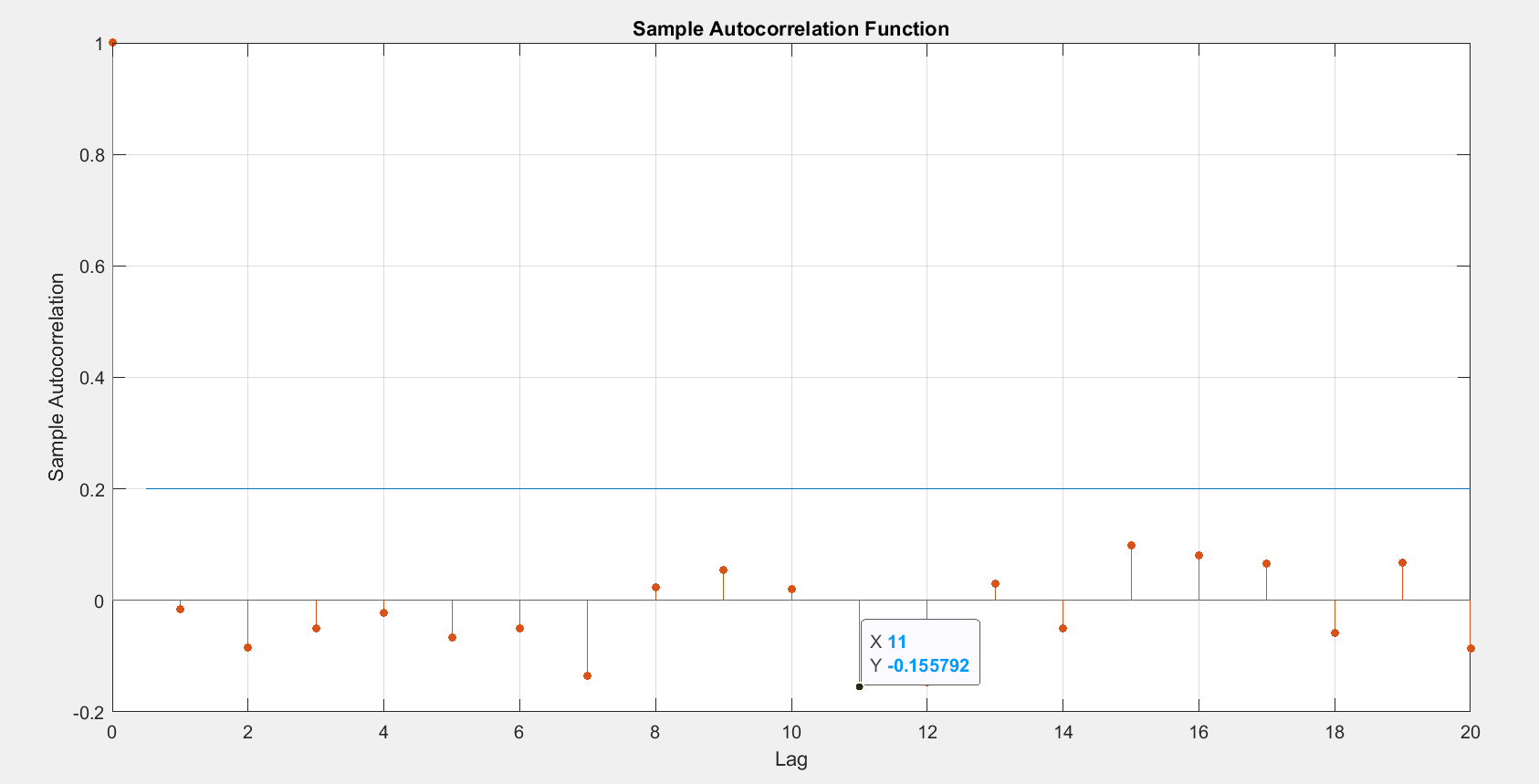
Number of batches:200 Length of batches:500

Number of batches:500 Length of batches:200

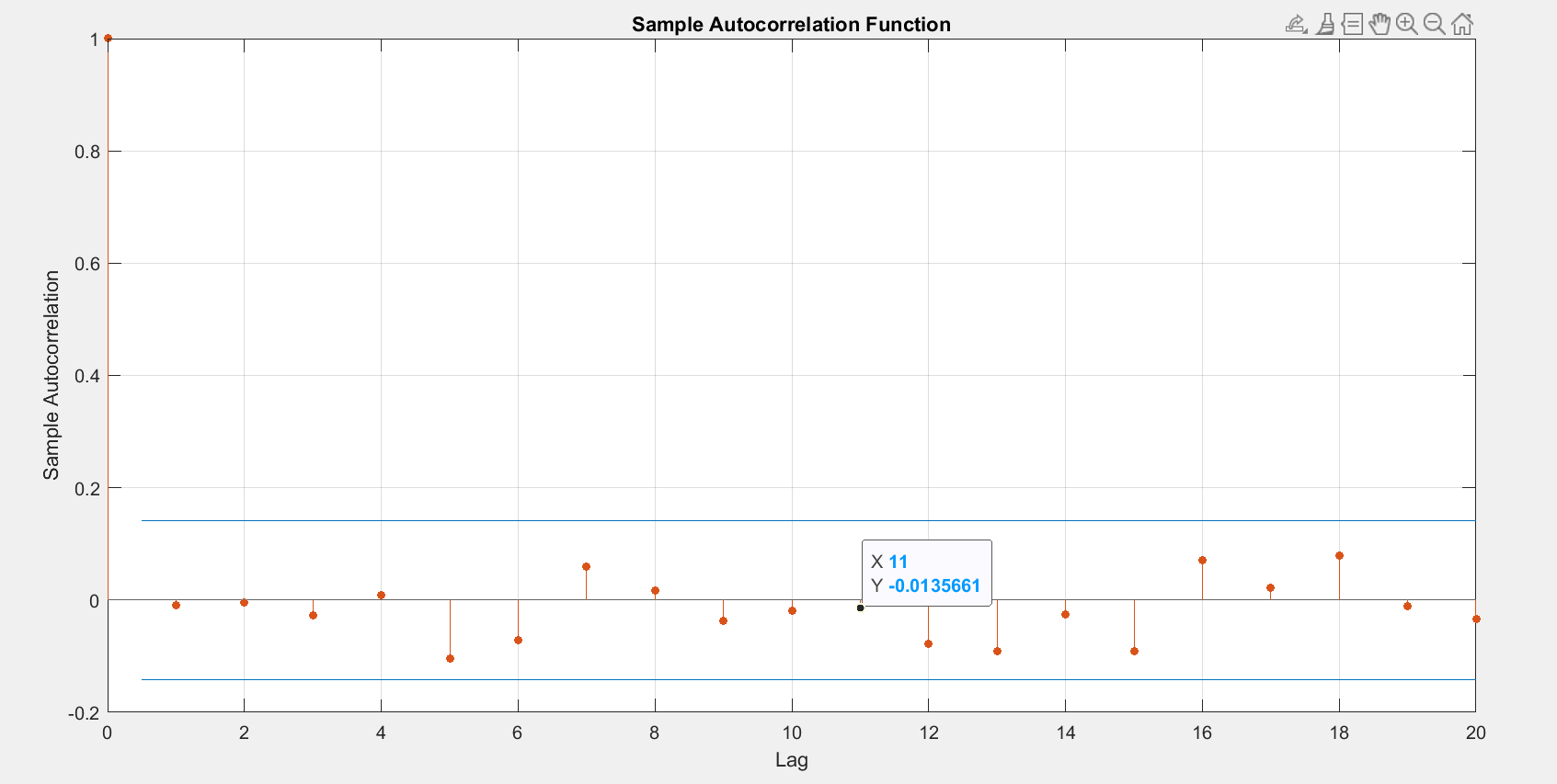
Number of batches:1000 Length of batches:100



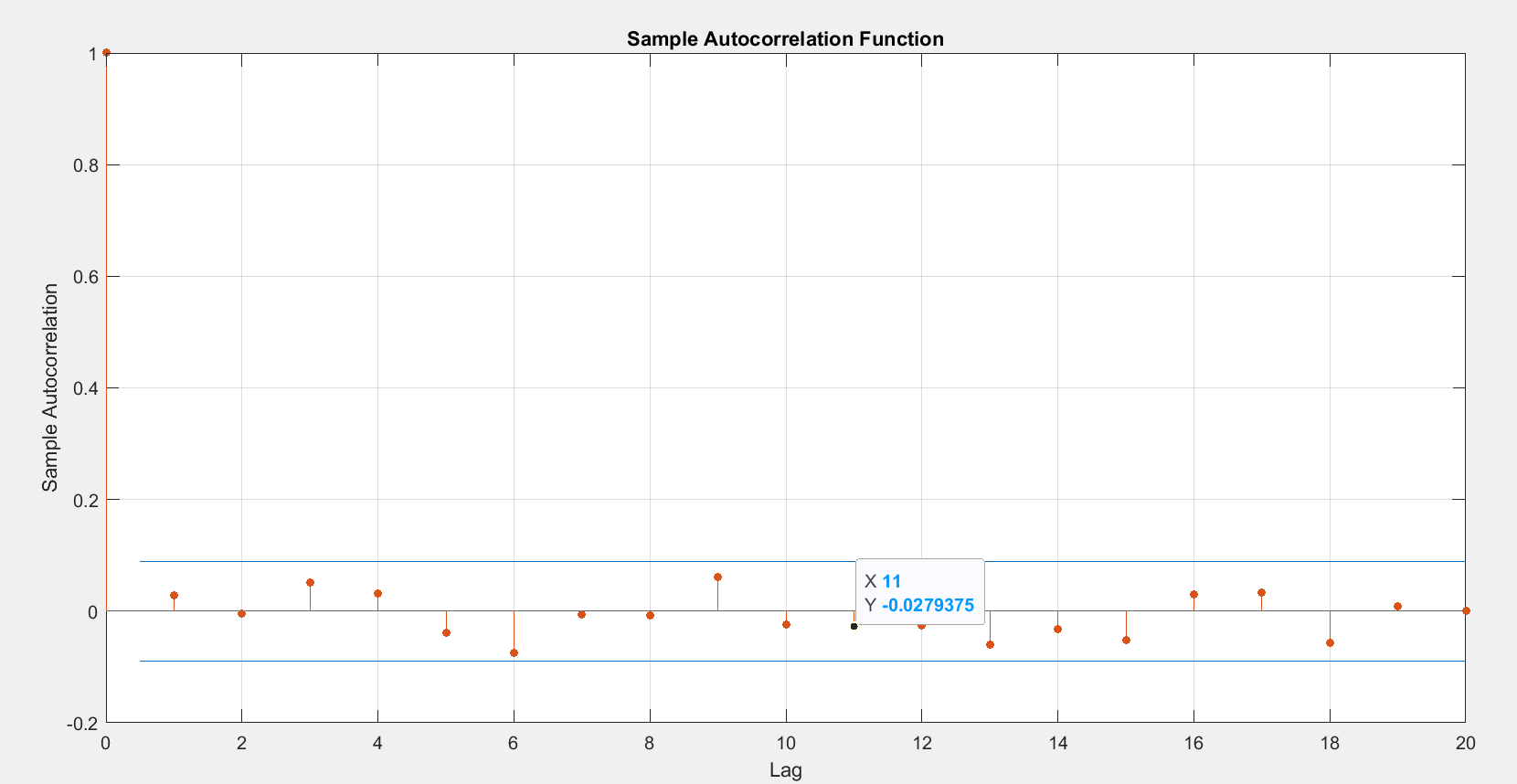
*Fig 6. 50 batches 2000 batches*



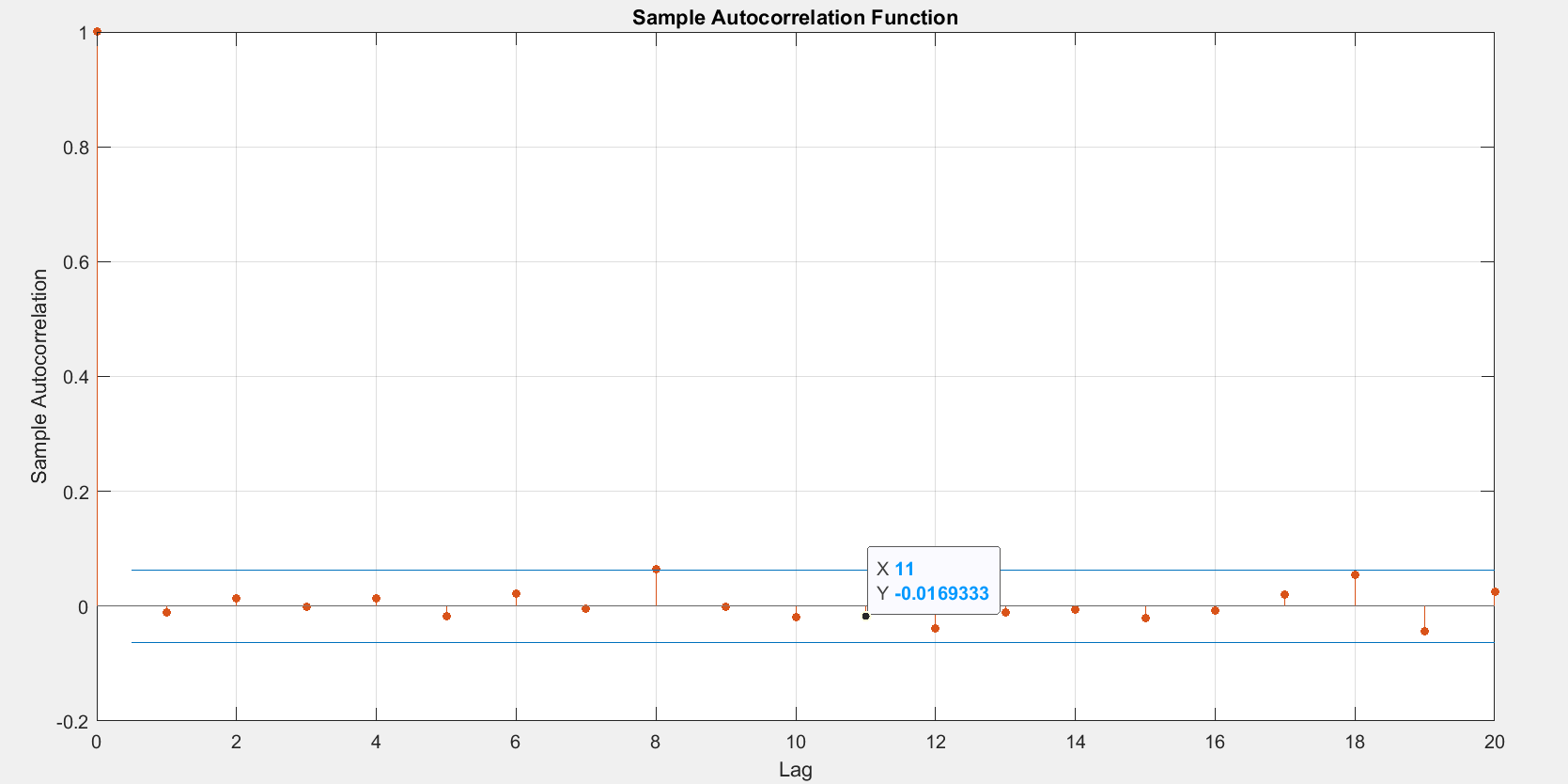
*Fig 7. 100 batches 1000 arrivals*



*Fig 8. 200 batches 500 arrivals*

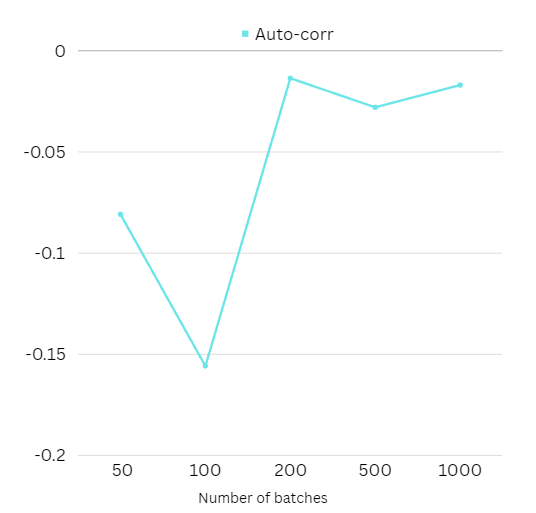


*Fig 9. 500 batches 200 arrivals*



*Fig10. 1000 batches 100 arrivals*

We simulated the system as above to check our **hypothesis** that as the data is random, **“*there will be no pattern*”** (i.e., the observations within the same batch may exhibit higher correlation compared to observations in different batches) as we increase or decrease the number of batches and length of batches what impact of varying the batches would do on autocorrelation, **“after running the simulations we found there was no significant impact weighing our hypothesis.”**



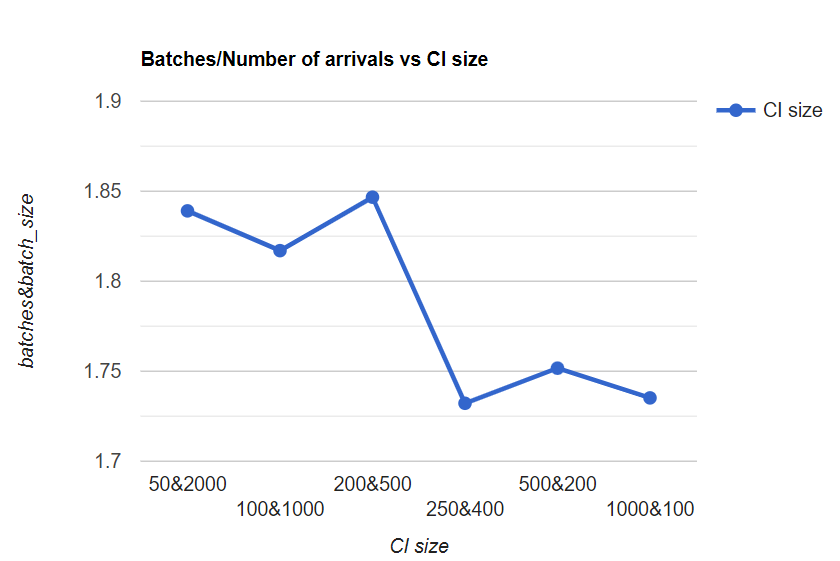
*Fig11 – Autocorrelation trends for varying batch length and sizes*

## **EXPERIMENT-3 (Q3,Q4,Q7,Q8,Q9,Q10)**

**IMPACT ON CONFIDENCE INTERVAL**

**Impact of the number of batches & length of batches on the size of the confidence intervals (Q3 & Q4)**

We hypothesize from our knowledge that “***variability in the data reduces as it attains steady state, therefore, we would here wish to validate that confidence interval size reduces as batches increases i.e., as number of arrivals reduce”***

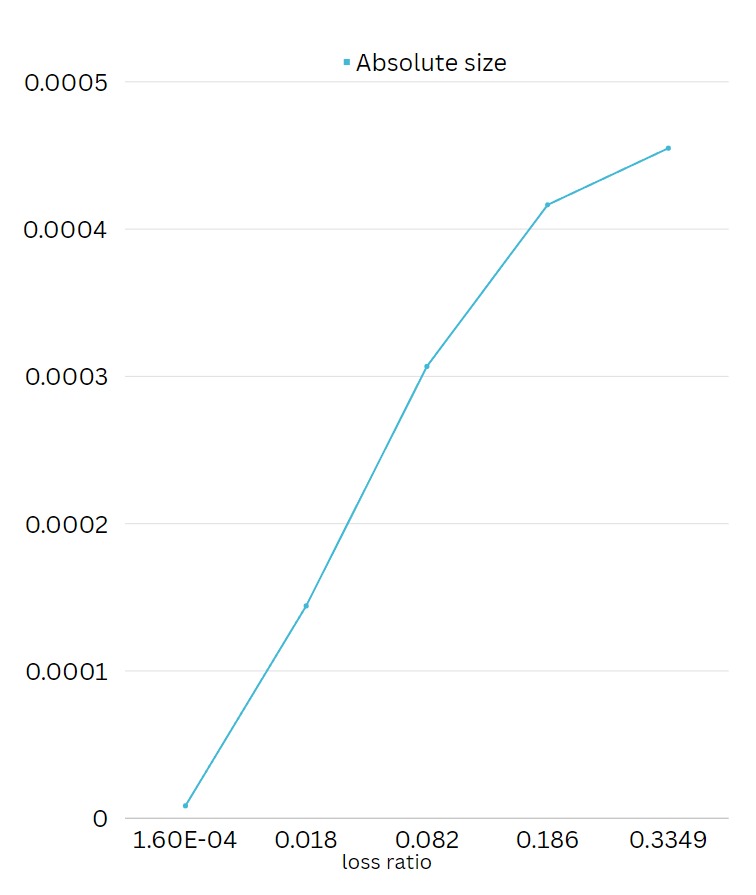


*fig 12 – CI size trends for varying number and length of batches*

**“From the results of simulation and observations, The values however reduce as expected with increase in number of batches and reduce in batch size, as evident in the fig 12.”**

**Impact of the size of loss ratio on the absolute sizes of the confidence intervals (Q7)**

Initially, to find the impact of loss ratio on absolute sizes, we had to make the loss ratios visible, so we made the service time greater than arrival time as follows  
Arrival time: 100  
Service Time:50,80,100,120,150  
***“By changing the service time we changed the size of loss ratio, As the Size of loss ratio varies the absolute size is expected to vary.”***

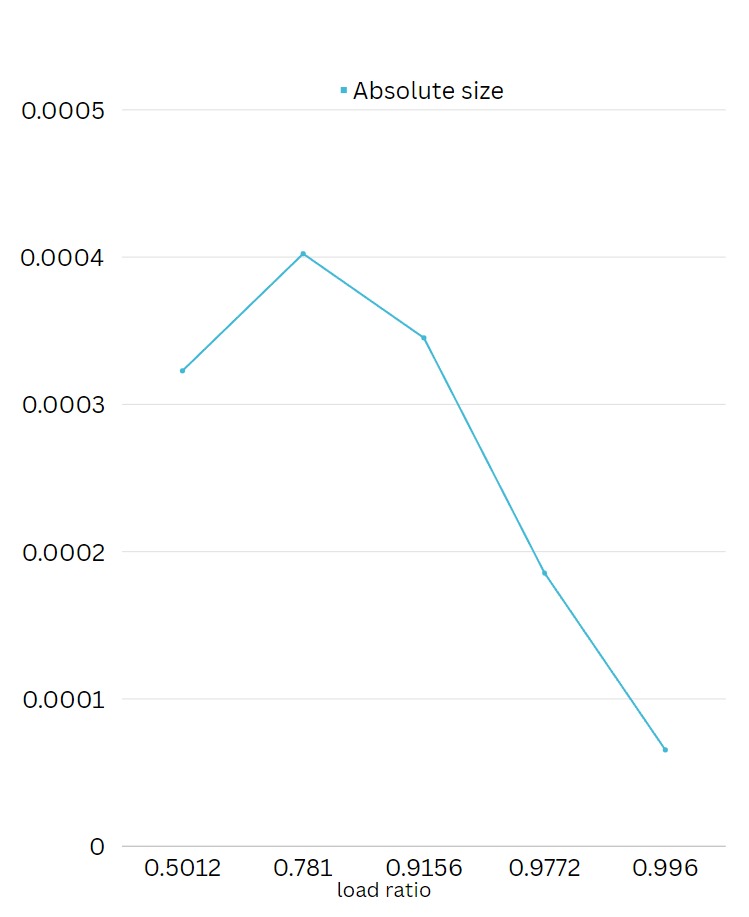


*Fig 13-Variation of Absolute Size due to varying size of loss ratio*

From the graph above we can see that **”The simulated data shows that absolute sizes keep increasing as the loss Ratio kept increasing”,** making our hypothesis of variation true.

**Impact of the size of load or queuing ratio on the absolute sizes of the confidence( Q8)**

Load ratio and Queuing Ratio are directly related, So the behavior of load Ratio will indefinitely show to the behavior of the queuing ratio.  
***“As the Load Ratio increases i.e., tends to ‘1’, the system starts to become unstable, therefore ultimately the absolute size reduces.”***   
By adjusting the parameters of system to increase the size of load ratio the effect of load ratio on the absolute sizes of CI is observed.

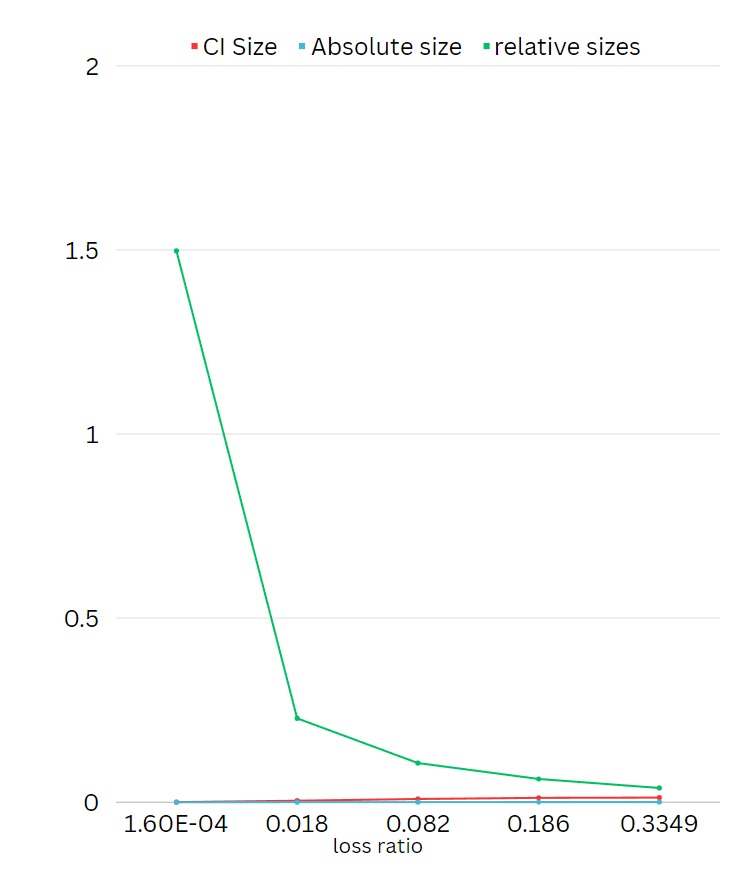


*Fig 14-Variation of Absolute Size due to varying size of load ratio*

**“From graph we can say that as the load ratio is nearing 1, the absolute size kept decreasing. For that matter, it validates that load ratio impacting the absolute size.”**

**Impact of the size of loss ratio on the relative sizes of the CI. (Q9)**

The service time here is adjusted to increase the size of the loss ratio and the values were as follows.  
Arrival time: 100  
Service Time:50,80,100,120,150  
By making the service time more than arrival time the loss ratio is made visible. “***The relative size of CI is inversely proportional to service time, i.e., it implies that loss ratio is also inversely proportional to relative sizes of CI***.”

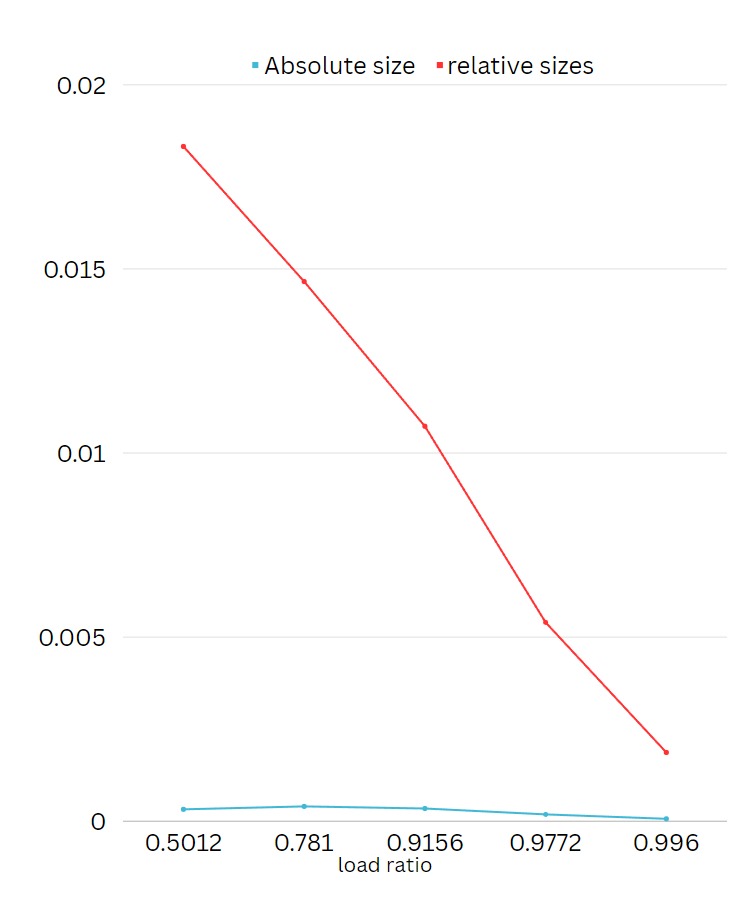


*Fig 15-Variation of relative Size due to varying size of loss ratio*

**From the graph derived from our simulated data shown above, we can verify that the relative size of the CI keeps decreasing as the loss ratio is increased, stated in our hypothesis.**

**Impact of the size of load or queuing ratio on the relative sizes of the confidence (Q10)  
intervals**

The queuing ratio depends on the service time, by varying the service time queuing ratio and load ratio can be varied, “***As the queuing ratio increases the number of people in the queue increases, eventually the load on the system increases and systems tends to be unstable.***”  
This will ultimately lead to reduction of the relative sizes of CI.



*Fig 16-Variation of Relative Size due to varying size of load ratio*

**From fig we can determine our hypothesis that as the load ratio is increasing the relative size increases.**

## **EXPERIMENT-4(Q5&Q6)**

**IMPACT OF SERVICE TIME**

**Impact of the average service time on load, loss ratio and queuing ratio (Q5)**

Service Time is the time taken for a customer to complete task, by varying the service time and compared with arrival times we can estimate the behavior of the system.

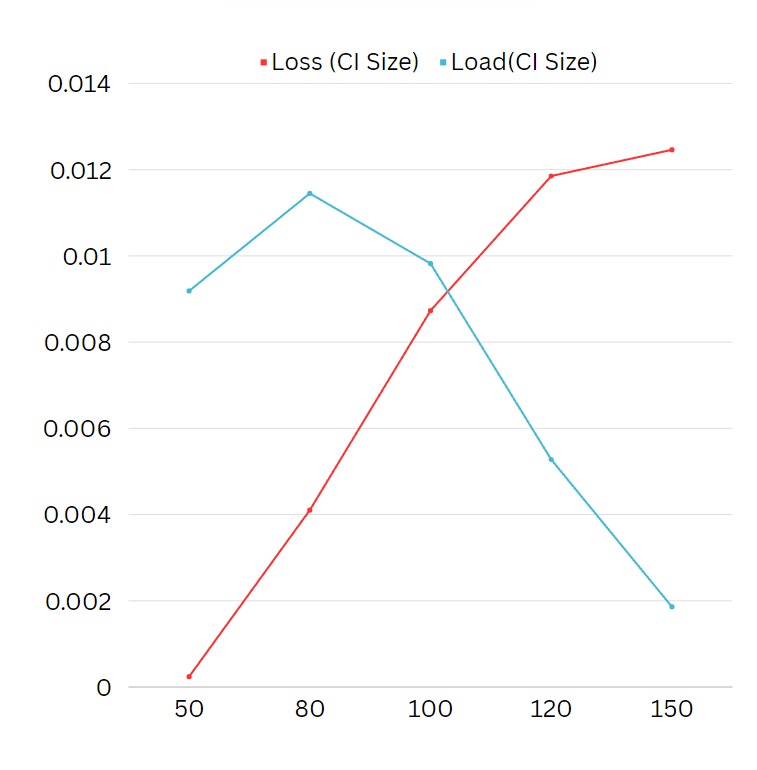
When service time increases and is nearing the Arrival time then the queue gradually increases as it takes more time to serve each customer, hence increasing the queuing ratio , this also increases the load on the system ,   
Therefore “***the load Ratio, loss ratio and Queuing Ratio increases as the service time increases***”.  
 Arrival Time: 100  
Service Times: 50,80,100,120,150



*Fig 17- Load ratio and loss ratio and Queuing ratio trends due to increasing service time*

The above figure shows the trends in load Ratio, Loss Ratio and Queuing Ratio for different Service Times. “**Load and loss ratios show increasing trend with increase in average service time validating the hypothesis given”.** But as the service time is increasing whereas the **Queuing Ratio shows an increasing trend until it reaches 100(i.e our arrival time), then it shows a variability as the system becomes unstable, this rejects our hypothesis of queuing ratio being directly proportional to the load ratio. Queuing ratio is indeed reducing just after unstable state is reached, compensating the load and loss values in it’s metrics.**

**Impact of the average service time on the size of the confidence intervals. (Q6)**We have taken the Confidence interval of load and loss as they exhibit different response to the change in the service time. *“****As the Service time increases, we expect an impact in load and loss ratios”.*** As the Service time increases the CI size of the load will be wider as the load will increase, but again when system gets overloaded the behavior of the system could be variable.



*Fig 18-CI size trends due to increasing service time.*

The same can be validated from fig17.

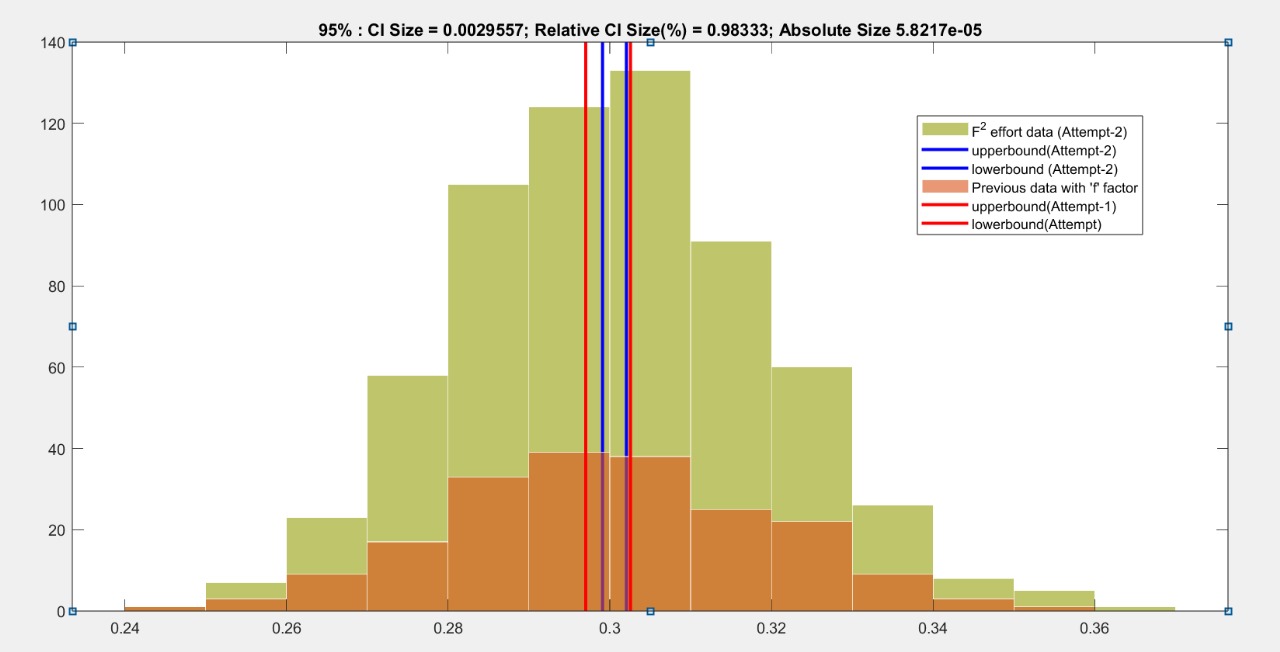
**“The CI of the loss ratio will behave a bit differently when compared to Load ,when the system gets overloaded the rejection of customers will increase thereby increasing the loss ratio, so the loss ratio will continue to grow. This can be seen from the figure 17.”**

## **EXPERIMENT-5(Q11)**

**F^2 RULE-OF-THUMB**

*The initial Relative size of confidence interval is ‘1.8%’ for ‘200’ batches and ‘500’ arrivals, achieved during experimentation of Q9.* “***Our aim is to reduce the CI even further, to achieve this, we have to multiply the number of batches to F^2 effort. (i.e., the value of ‘f’ or effort here is the relative size of the CI)”,***By doing this F^2 rule we increase the total sample size, which will reduce the relative size of CI. In our case less than 1%. OUR F-FACTOR WILL THUS BE (PREVIOUS/DESIRED) = (1.8/1)= (1.8).

**“As desired, less than 1% CI is justified in the results after the simulation.”**



*fig 19-f2 Rule*

The idea behind this rule is that as you increase the number of batches, the size of the confidence interval decreases, providing more precision in the estimation. The reduction is proportional to the square root of the number of batches. This has been validated as shown in the figure, the confidence interval which has been 1.8% relative size has bene reduced to less than 1% as desired, with an effort of (1.8)^2

## **EXPERIMENT-6(Q12)**

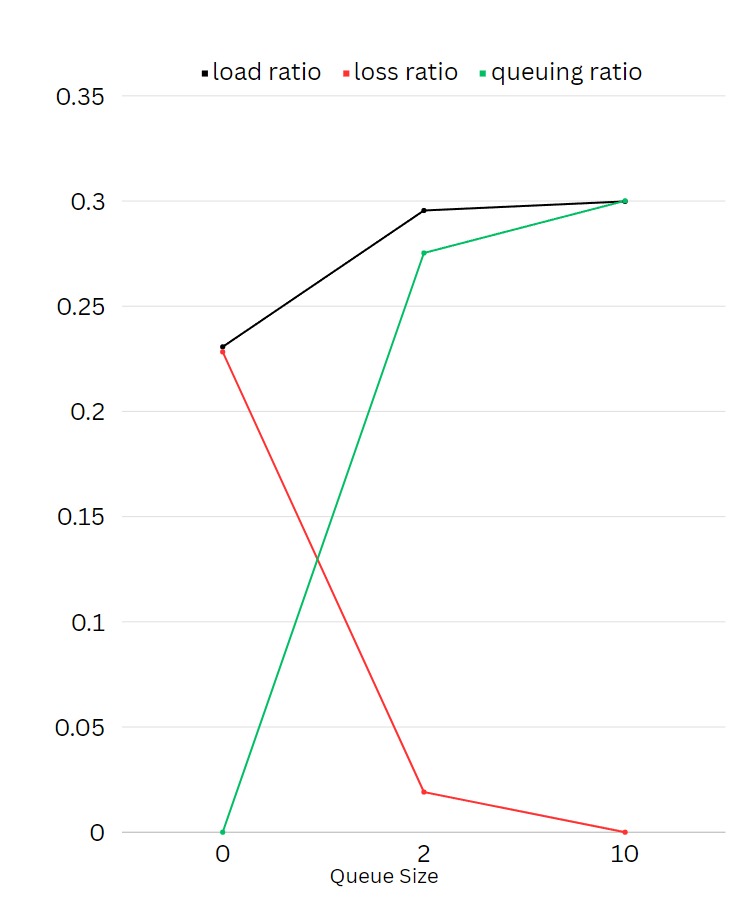
**IMPACT OF DIFFERENT QUEUE SIZE**

***“Changing the queue size have a huge impact on the system as the size of the queue has a direct impact on load, loss and queuing ratio of the system.”***

***“Load will be less when there is no queue***” but no zero as load is affected by many different factors**. *“As the queue increases the load will increase.”***

Loss Ratio has an opposite behavior when compared to Load Ratio, the reason for this is if there is no queue the upcoming customers will have no place to be in the system. When the system is busy all the new customers arrived will be dropped as soon as they arrive.  
Now “***as the queue size increase, there is place is for more and more customers, so the rejection will decrease , which reduces the loss in the system***.”

Initially when the queue is zero the queuing ratio will be zero as there are no people in the queue. *“****As the queue size increases the number of people in the queue will increase.”*** The queuing ratio will increase as the queue size increases.



*fig 20-Queue size effect on load, loss and queuing ratio*

**“From the above figure(fig 19) we can observe that load and Queuing ratio are increasing .Queuing ratio starts from zero and then increase but load increases from a value validating our hypothesis. Also the loss Ratio has a decrease in its value as the queue size increases as stated in our hypothesis.”**

## **EXPERIMENT-7(Q13,Q14,Q15,Q16)**

## **IMPACT OF LOAD AND LOSS RATIO IN A/B/C SYSTEMS**

***“Randomness in the server and source process impacts to increase the loss ratio and queuing ratio”***

## While number of parallel servers is 1, to compare the loss and queueing ratios, while making the values visible, we considered the input values in 2 cases

## **Case:1** – By adjusting desired load > 1

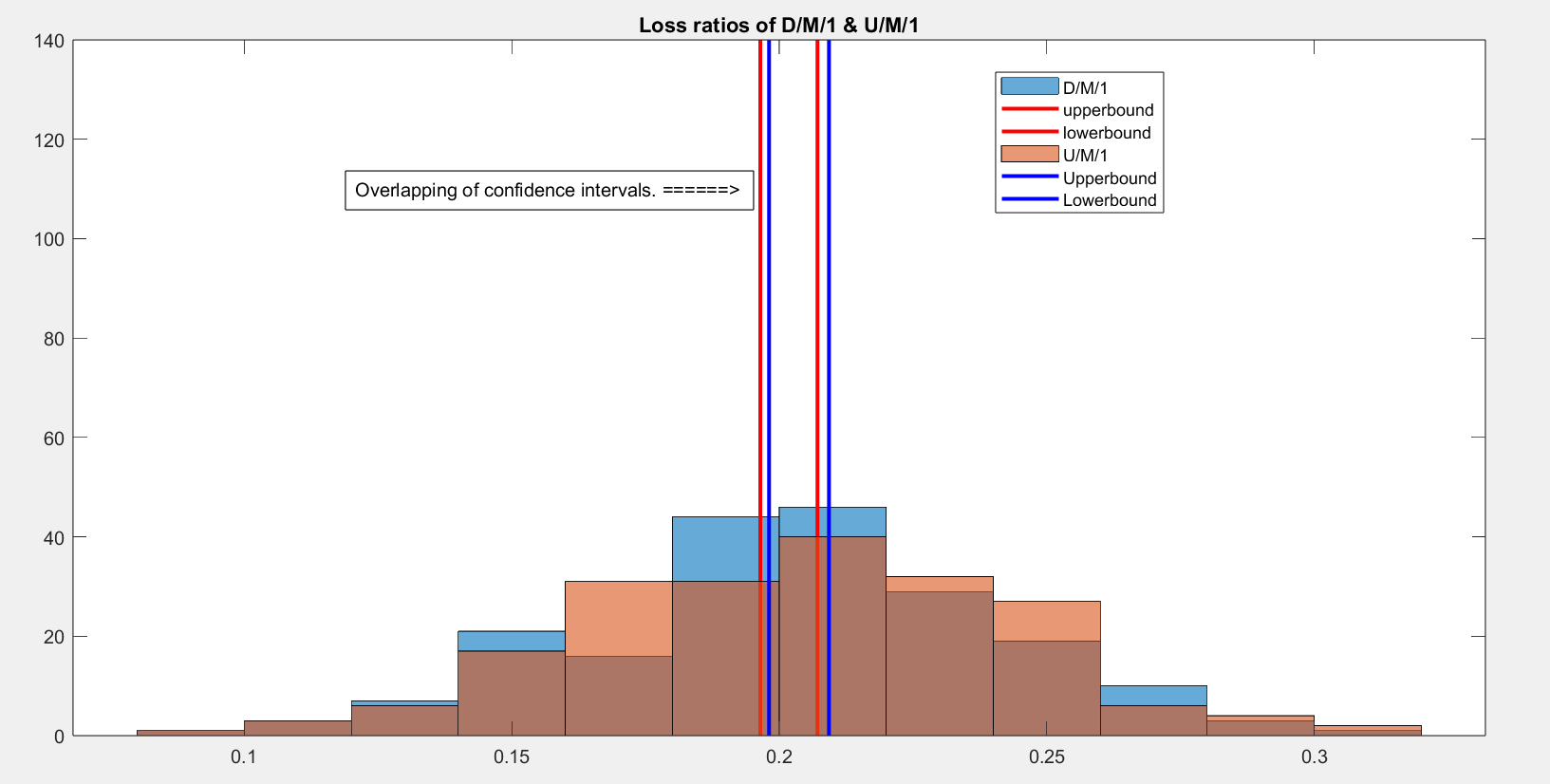
## Arriving = 80

## Servung time = 100

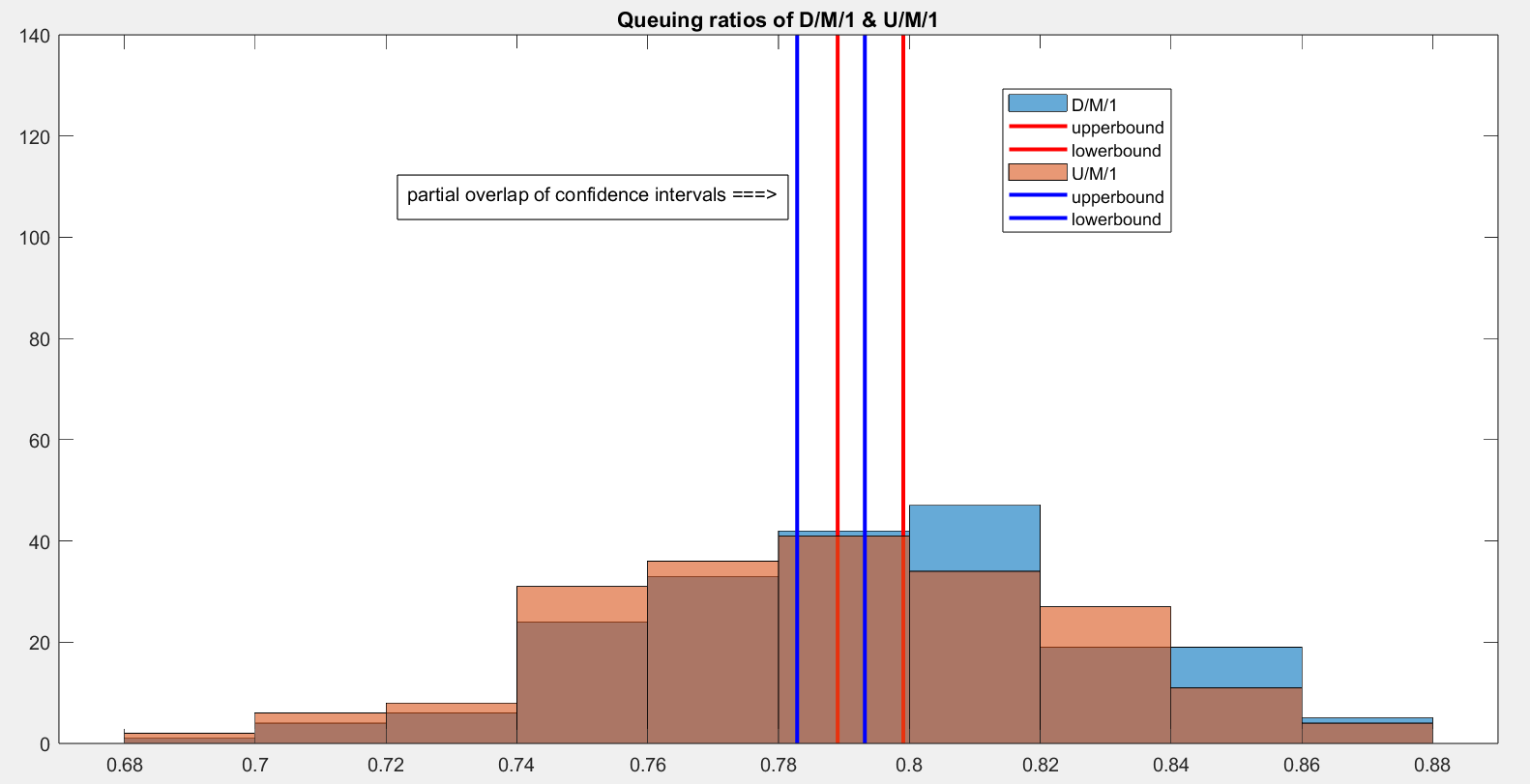
## **Case: 2** – By considering lower queue sizes as in k=2

## **1)loss and Queuing ratios of D/M/1 &U/M1**

## **Case:1** – By adjusting desired load > 1



*fig 21- Loss ratios of D/M/1 & U/M/1*



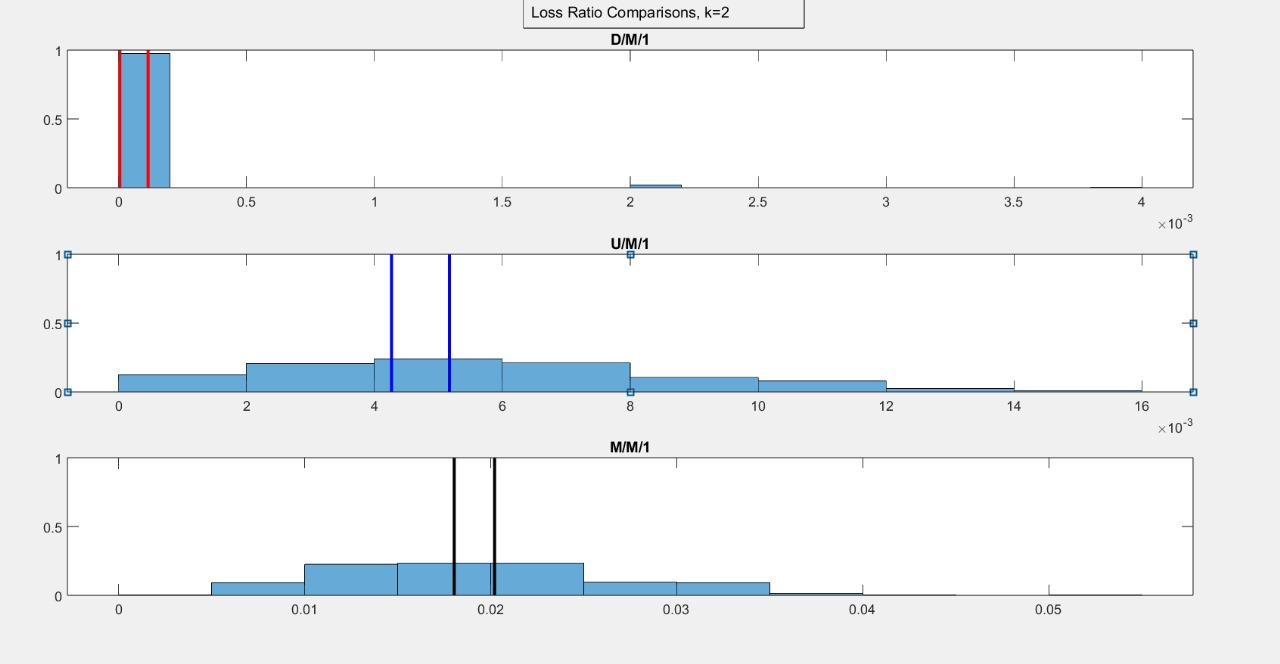
*fig 22-Queuing ratios of D/M/1 & U/M/1*

The risk of overlapping in the datasets:  
When the CIs of μ1 and μ2 overlap, it implies that the range of values considered plausible for each parameter includes portions of the other's range. This overlap makes it challenging to confidently conclude that there is a true difference between μ1 and μ2. Overlapping CIs indicate a level of uncertainty, and it becomes difficult to claim with high confidence that the two values are distinct. There is a risk of ambiguity in asserting a significant difference.

If the CIs do not overlap, it suggests that the range of plausible values for μ1 and μ2 is distinct. This scenario provides more confidence in stating that there is a significant difference between the two p Non-overlapping CIs reduce the risk of ambiguity, especially when using a higher confidence level (CL). A higher CL implies a wider CI, but if the CIs don't overlap, the risk of misinterpretation diminishes.

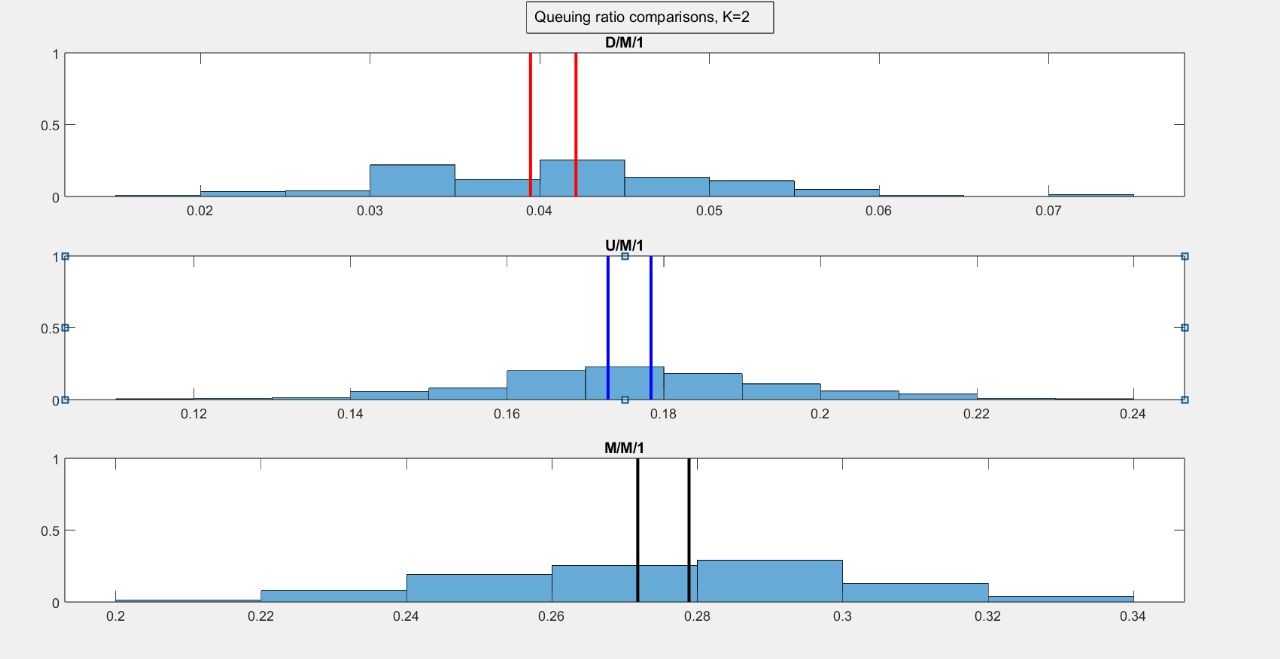
We have chosen to first go with reducing queue size to see the behaviors of datasets in overlapping confidence intervals. In case of overlapping of CI’s during this case, we would have proceeded with reducing the size of CI’s using f^2 thumb rule. But, incidentally, distinguished datasets have been produced when queue size was reduced.

Case-2: limiting queue size, k=2



*fig 23-Loss ratios, when k=2*

Loss ratios are reduced in D/M/1 and U/M/1 systems compared to M/M/1 systems as randomness reduced.



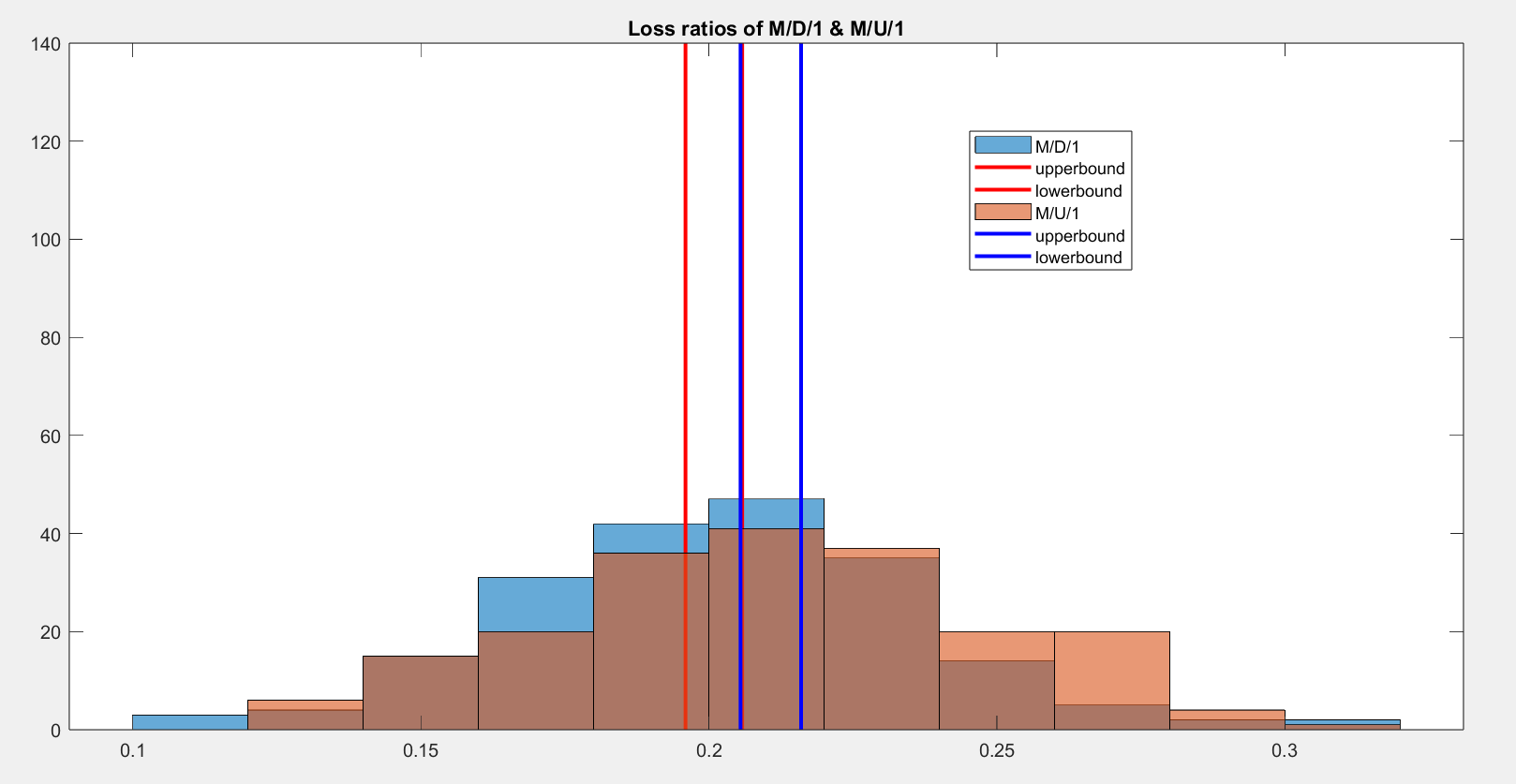
*fig 24 – Queuing Ratios when k=2*

**“The distinguished datasets, and this reduces ambiguity to comment and validate the behavior. The impact of the randomness of source processes is evident queueing ratios of D/M/1 and U/M/1 are lesser than M/M/1 queuing ratio.**

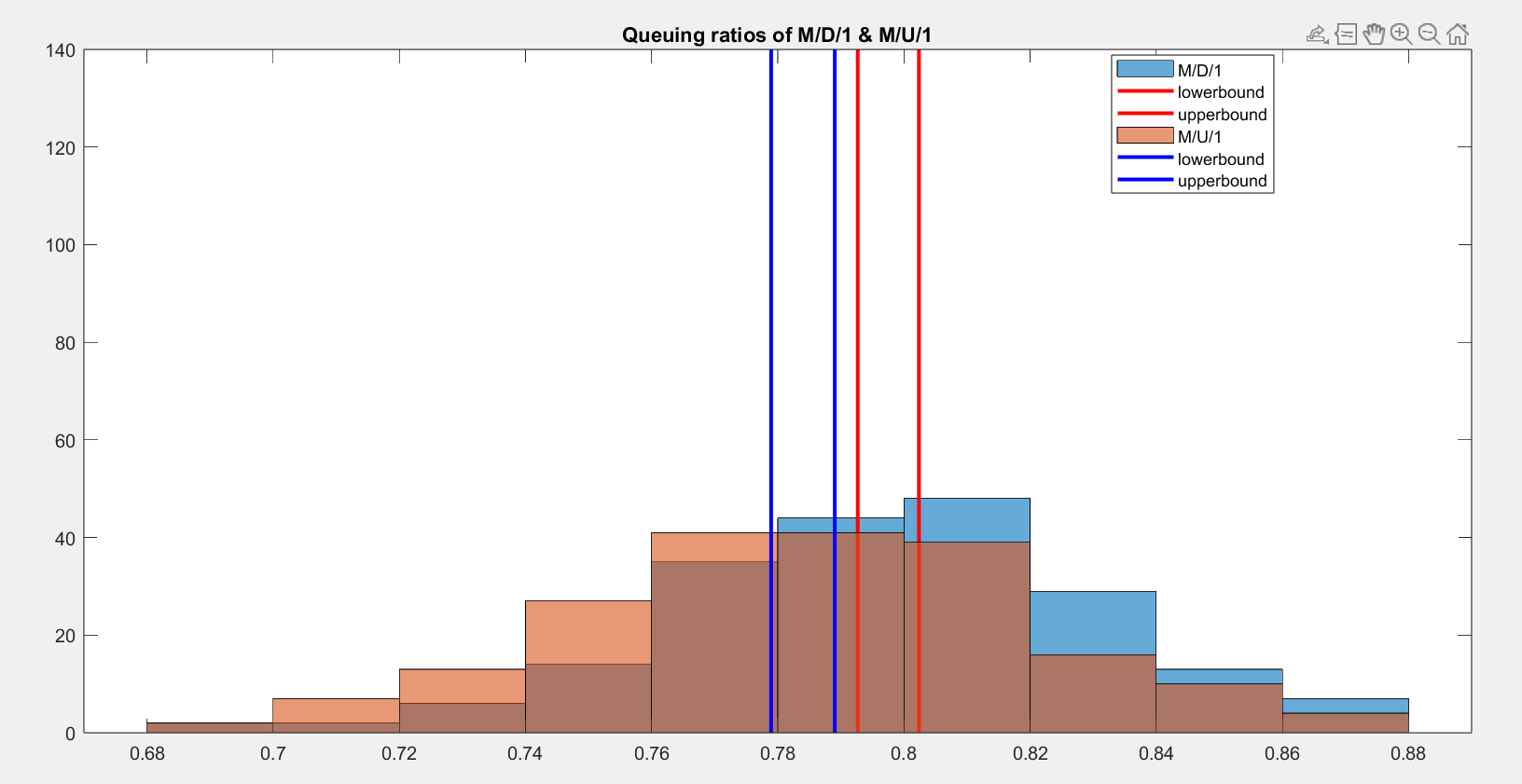
**Interestingly, queuing ratios are almost similar in all cases around 0.28 to 0.3, as also seen in fig-23. Irrespective of server processes being changed, the values of means of queuing ratio remained still”**

2)loss and queuing ratios of M/U/1 & M/D/1

Case:1 – By adjusting desired load > 1



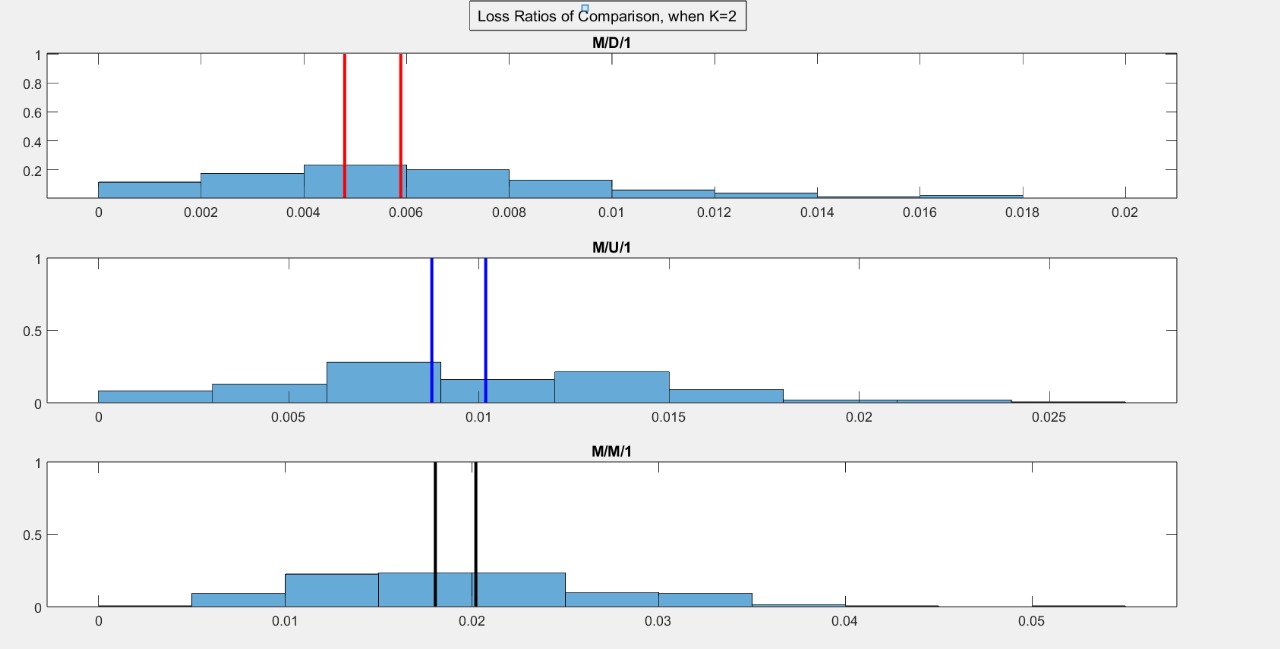
*fig 25-Loss ratios of M/D/1 & M/U/1*



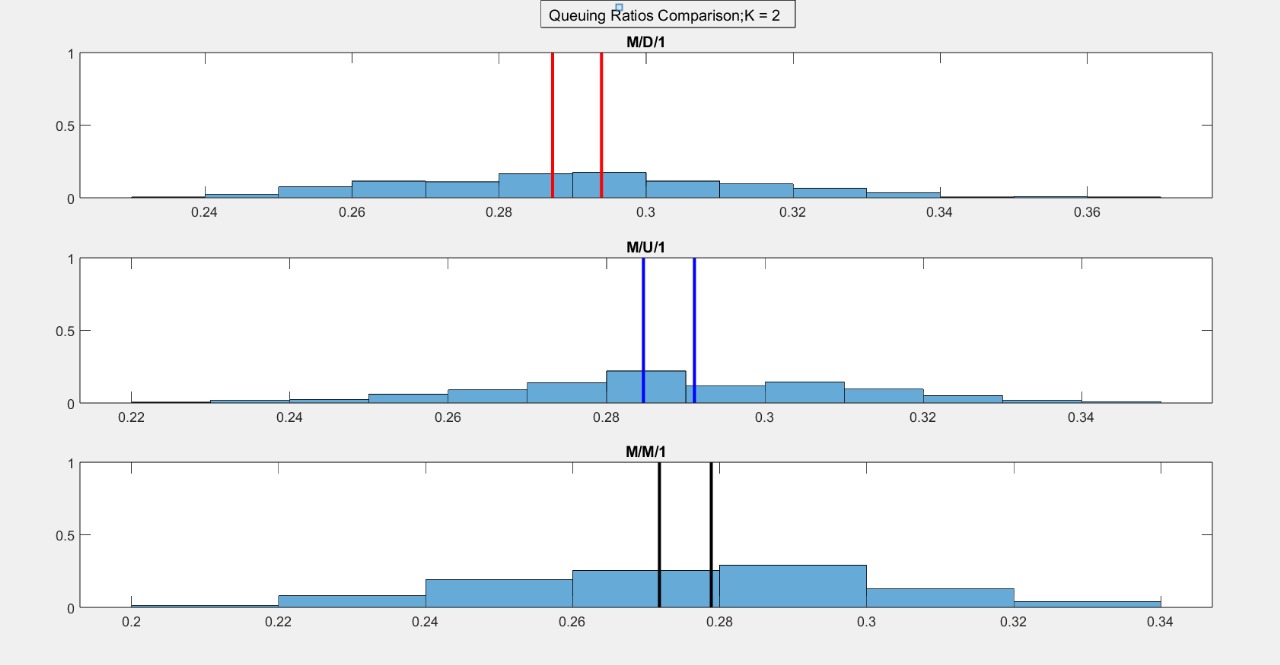
*fig 26-Queuing Ratios of M/D/1 & M/U/1*

**“Loss ratios and queuing ratios are not only less but are almost similar in M/D/1 & M/U/1. When server processes are fixed, the loss ratios are within 0.5% interval size, while the queuing ratios are within 2%. This validates that reduction of randomness in source or server process makes the performance ratios to be lesser than markovian source and server processes.”**

Case 2: Limiting Queue size, k =2



*fig 27- Loss Ratio when k=2*



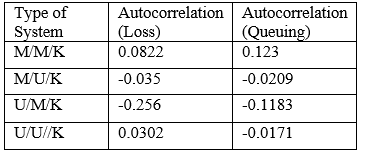
*fig28- Queuing Ratio when k=2*

**“On the whole, to compare the arrival and server processes when deterministic or uniform distribution has been considered, loss ratios have been extremely low compared to M/M/1. This validates that reduction of loss, when randomness is reduced. D/D/K is hence trivial, as loss ratios would be zero, when there is no randomness”**.

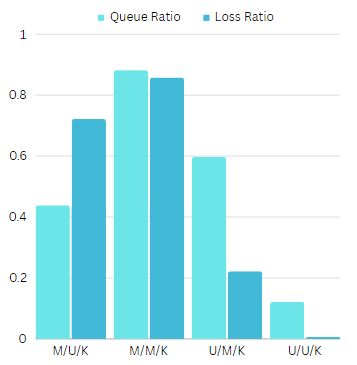
## **EXPERIMENT-8(Q17,Q18,Q19) – MOTIVATION OF OBSERVATIONS**

## **1) Loss Ratio And Queuing Ratio Of M/M/K With Respect To M/U/K, U/M/K & U/U/K Systems(Q18,19)**

We have written a code with single queue multiple servers and where the customer in the queue goes to the least busy(or least queue length) server and have determined autocorrelation of the systems to determine the state of the systems, to uncorrelated in their loss and queuing ratio initially then proceeded with observing loss and queuing ratios. 80% is the desired load input.

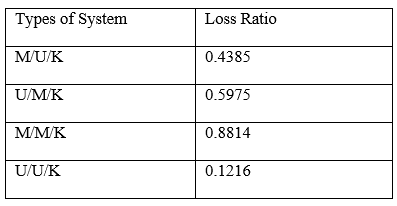


*Table1:Autocorrelation of M/M/K, M/U/K,U/M/K,U/U/K systems*

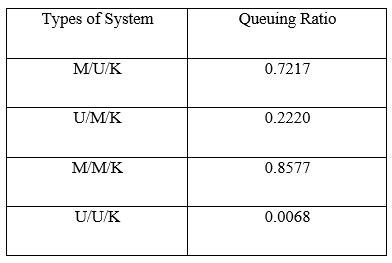


*fig. 29 Queuing and loss ratios of M/U/K, M/M/K, U/M/K and U/U/K systems*

**Motivation of observations:** **A memoryless model made the system perceive high queuing ratios and loss ratios, in parallel servers as well, although performing better than single server. Also, it is however known that, parallel servers makes the system efficient. The uniform randomness (in server process as well source processes) reduced the queuing and loss ratios, than in the markovian sources or server processes. U/U/K is giving extremely good loss and queue ratios.**

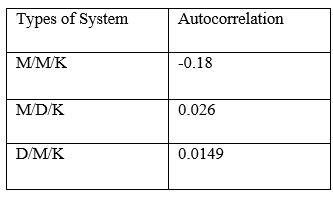


*Table2 : Loss Ratio* *of M/M/K, M/U/K,U/M/K,U/U/K systems*

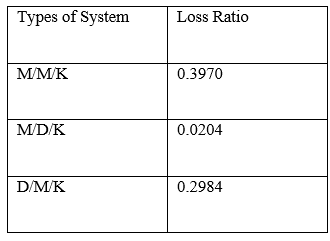


*Table3 : Queuing Ratio of M/M/K, M/U/K,U/M/K,U/U/K systems*

**Loss Ratios For The M/M/K’’=10 System With Those Obtained For The M/D/K’’ And D/M/K’’ Systems At 80 % Load, And Motivate Your Observations (Q17)**



*Table 1: Autocorrelation of M/M/K, D/D/K,D/M/K systems*



*Table2 : Loss Ratio of M/M/K, D/D/K,D/M/K systems*

Even when the parallel servers are increased, the randomness in source and server process, increases the loss ratio as stated in our previous discussions and hypothesis.

**The observations motivate us to validate deterministic behavior in server process reduces the loss ratio to ultimate extent than M/M/K” and D/M/K”, although “Deterministic” behavior in sources or server is however smaller than “Markovian source and Markovian server”**