

# A Case on Queue-Systems Analytical and Simulation Solutions

## By SAN Formalism

Mohammad Reza Besharati<sup>1</sup>, [besharati@ce.sharif.edu](mailto:besharati@ce.sharif.edu)

Mohammad Izadi<sup>1</sup>

1. Sharif University of Technology, Tehran, Iran

### **Abstract**

There is a case about a system comprising two queues of  $M/M(a)/1/K$  with a due date with FCFS service policy in each queue. Using the methods of simulation and mathematical analysis, the solution diagrams were illustrated for the fixed due date  $\Theta=1$  second at the phase of due date up to the commencement of the service, limited capacity  $K_1=5$  and  $K_2=4$ , for both scenarios of different service management rate resulted. For the analytical solution, first, a SAN model from the system was created in the Mobius tool. For simulation, a java discrete systems simulation code was used. Adjustment of the initial values, selection of the scenarios for service rate, the number of the consumers in the simulation, etc. were all carried out in the simulation code.

**Keywords:** Queue-Systems, SAN model, Mobius Tool, Simulation

### **System Definition**

There is a system comprising two queues of  $M/M(a)/1/K$  [1] with a due date with FCFS service policy in each queue. A customer logs into the system, in case the number of the costumers waiting in both queues are the same, the customers enter either queues with the same probability. Otherwise, they will wait in a queue with fewer customers. Only in case, both queues are full, the customer will be blocked. The service management rate by each queue is a function of the number of customers in that queue. Here, the due date of the customer is merely up to the time of commencement of the service appointment, and after that, they can remain in the system up to the end of receiving the services.

Simulation is an important system design method [4]. Using the methods of simulation and mathematical analysis, the following diagrams were illustrated for the fixed due date  $\Theta=1$  second at the phase of due date up to the commencement of the service, limited capacity  $K_1=5$  and  $K_2=4$ , for both scenarios of different service management rate resulted from the following functions.

- For each queue, the diagram of the costumers that lost their due date ( $X_d$ ) in relation to the changes of the input rate of the system  $\lambda$  (in the range of [0, 1-20] with the churn rate of 0.1) was illustrated and the average simulation error was calculated in relation to the mathematical analysis.
- The diagram of the percentage of the costumers that fill the system upon logging in and are blocked ( $X_b$ ) in relation to the changes of the input rate of the system  $\lambda$  (in the range of [0, 1-20] with the churn rate of 0.1) was illustrated and the average simulation error was calculated in relation to the mathematical analysis

For a precise simulation, the number of the costumers at every input rate must be a large number (e.g. 1000000 costumers) which was observed.

Two different scenarios were considered for the values of the service management rate of the queues:

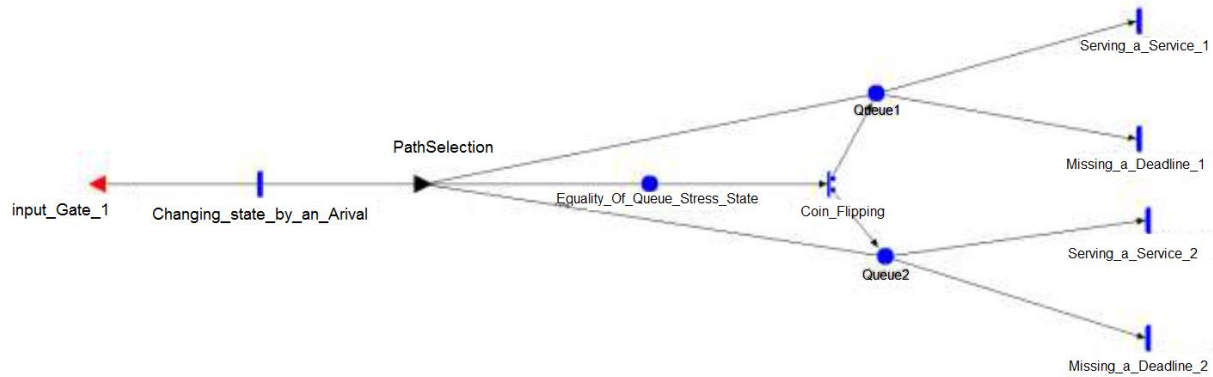
$$1) \quad \mu_1(n)=0.2n, \mu_2(n)=\text{Log}(n+1)$$

$$2) \quad \mu_1(n)=\frac{\sqrt{n}+n}{10}, \mu_2(n)=\frac{\sqrt{n}+n}{10}$$

## **the Process of the Proposed Solutions**

### **Analytical Solution**

For the analytical solution, first, a SAN model [2] from the system was created in the Mobius tool [5], [6], [7]. To do so, a section of the documentation of the Mobius that explained SAN Primitives was employed. The following figure includes the final model.



**Diagram 1 – SAN model for this system, this figure was created through Document Model from the File menu at Mobius tool.**

Failure to use ; in the functions in the modeling environment, incompatibility of the functions defined in the Header File and calling them in the modeling environment, failure to observe the SAN Primitive logic in connecting the components during modeling, etc. were among the issues that failure to notice them resulted in errors.

Another challenge was the concept of Precondition for an Activity defined by Input\_Gate. Finally, the following evidence was achieved: considering that each Activity is a Transaction and a change in the system State, it must be noted that every change in the State might have a precondition. Therefore, we put this precondition, if existed, in Input\_Gate.

If in the available Input\_Gate in the above diagram, the precondition of occurrence of changing\_state\_by\_an\_arrival, which is empty spaces in the queues is not considered, Absorbing State will be expected. Besides, the general variables shown in the Figure hereinbelow are defined in the model. The mioType means selecting the first or second scenario for the service functions that both scenarios are introduced earlier in the statement of the project.

Name	Type
Queue1_MaxSize	short
Queue2_MaxSize	short
Lambda	double
mioType	short

**Figure 1- General Variables of SAN Model of this System**

Then, the reward was developed for this system, which comprised three variables for total blocking, expiration of due date of queue 1, and expiration of due date of queue 2. The Reward calculation functions are inside Header File and are called from inside of Reward. In the Time section, Steady State was selected for all three variables.

Thus, I determined the Study. Assigned values to Lambda through Manual Range and by Import from a txt file. Since the Incremental Range creates the decimals for Lambda. With regard to the 200 different Lambda rates and two scenarios of different service rates (1 and 2 mioTypes), 400 tests were defined.

I produced State Space in the Transformer section. An error pertinent to the link was occurred and executed the order in the website for transferring the libm.a file in Cygwin.

Through the Iterative type of Solver, the analytical solution of the models resulted from Transformation was carried out and the results were read and extracted through a small assistant application, i.e. developed by Java, from the 400 files provided from Solver.

### **Simulation**

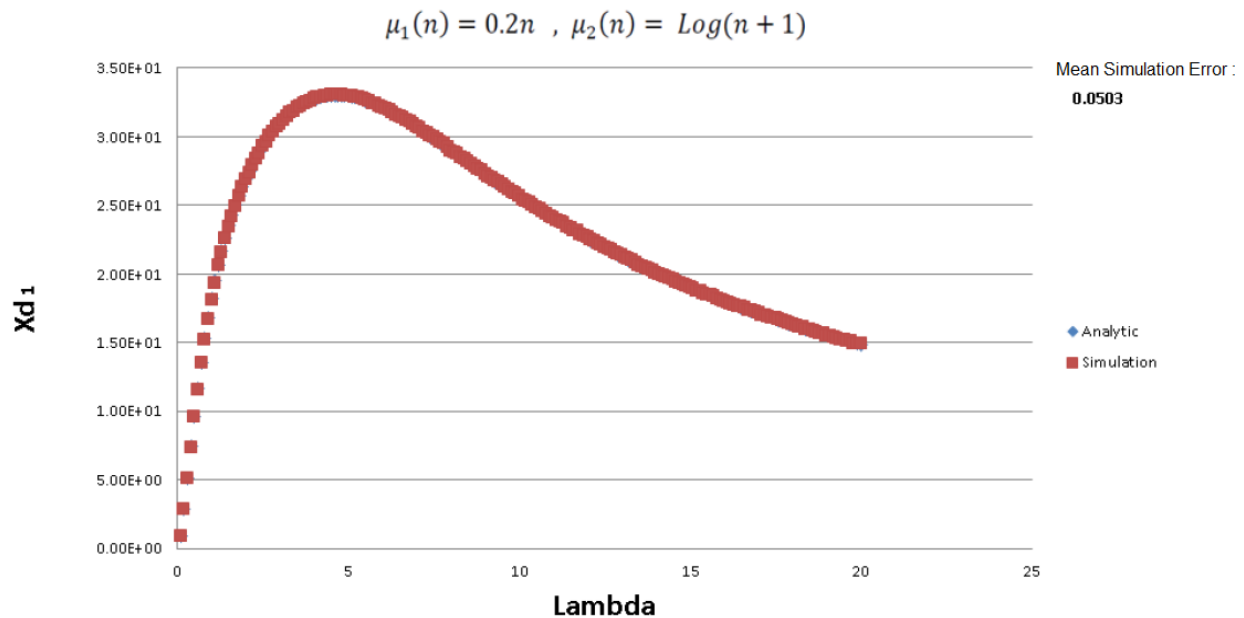
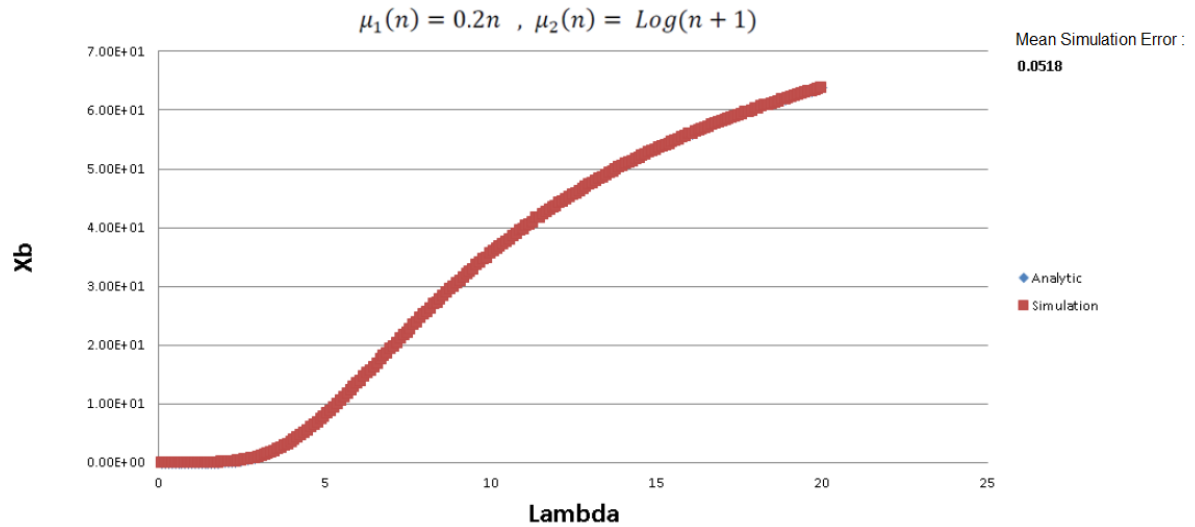
For simulation, a java discrete systems simulation code [3] was Refactor to obtain an object from the Queue class. Then, two objects were constructed from the Queue class at the main function from the Main class. Therefore, for the execution of the simulation, the main function from the Main class must be executed. Adjustment of the initial values, selection of the scenarios for service rate, the number of the consumers in the simulation, etc. were all carried out in the main Function at the Main class.

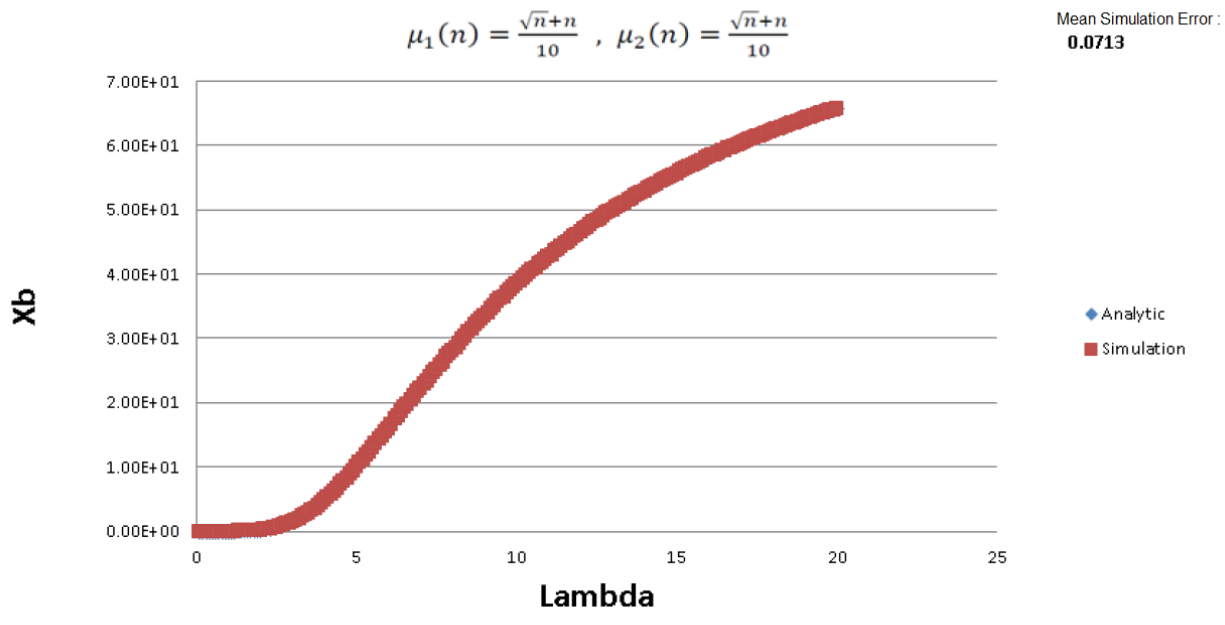
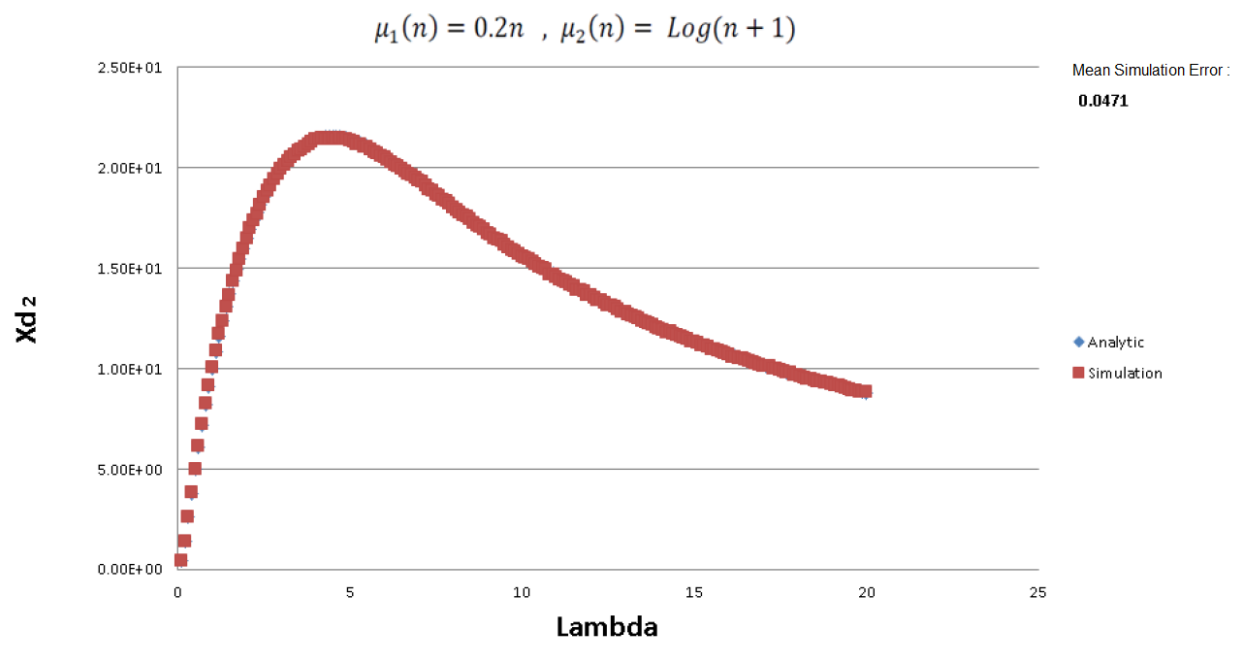
### **Error Average Calculation**

Assume that per input rate of  $\lambda_i$ , the analytical solution of the A value for the variable is calculated. (For instance, 65.7 percent) and per simulation, the value of B for the respective variable is calculated (For instance, 65.68 percent).

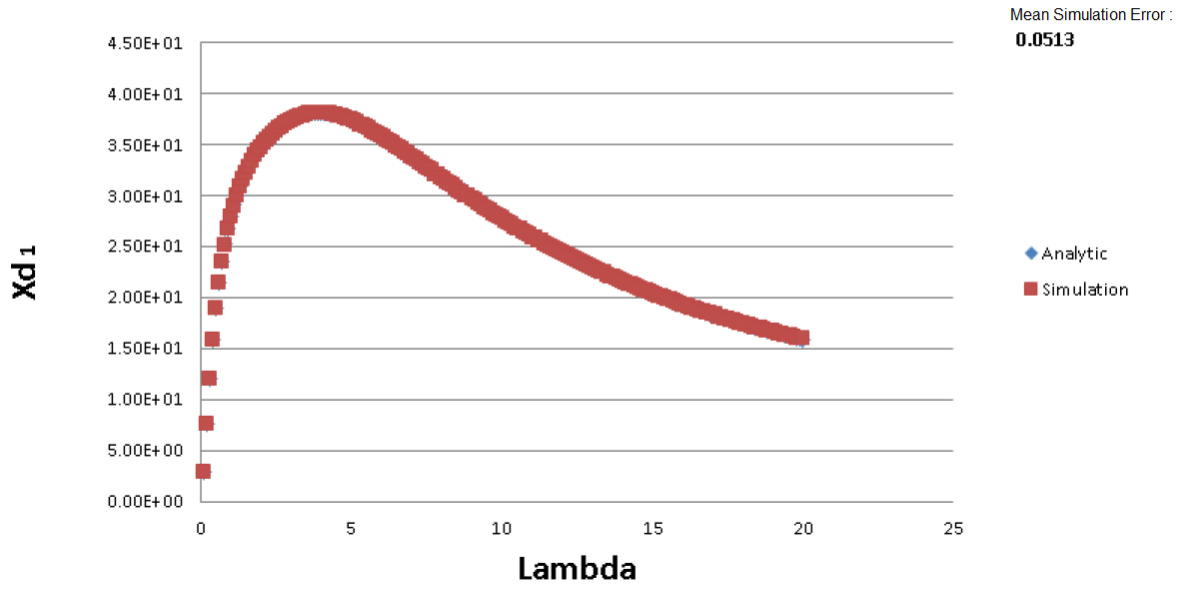
In this case, the absolute value of the difference between A and B (in this instance, 0.02) is calculated. Thus, this will be carried out per all values of  $\lambda_i$  and the average of all these errors is calculated.

In case the values of A and B are calculated as the probability (and not in percentage, i.e. between 0 and 1 and not from 0 to 100), the error order will be from the order 10 to the negative four power. However, now that the values of A and B are calculated in percentage (between 0 and 100) and not probability, the error order is from the order 10 to the negative two power. The average error values are mentioned next to the diagrams.

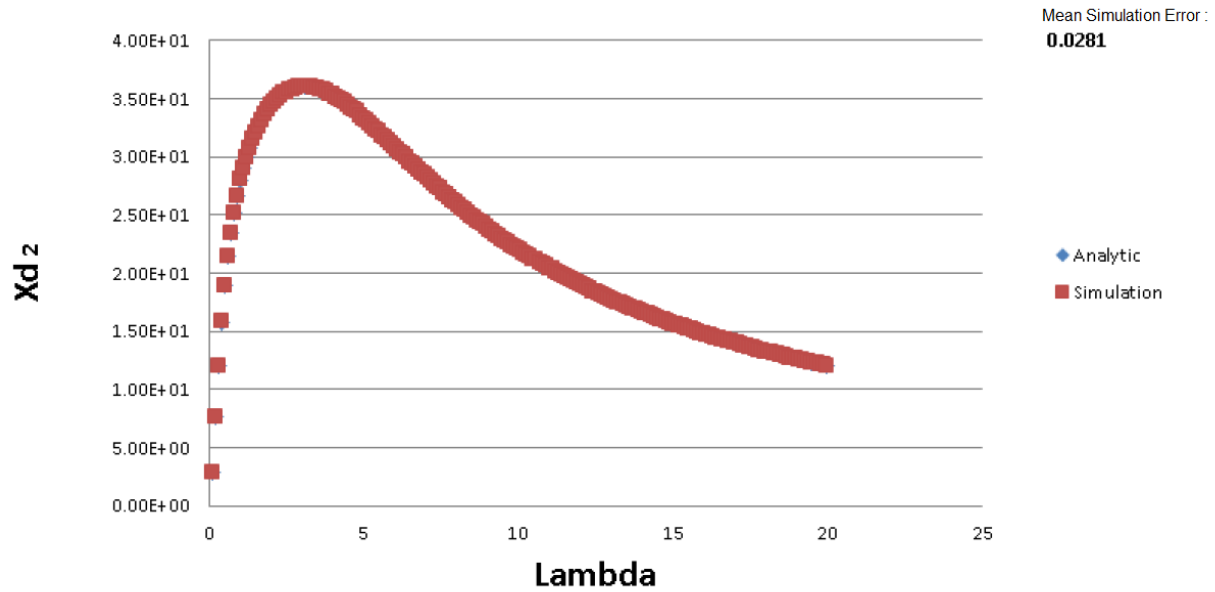


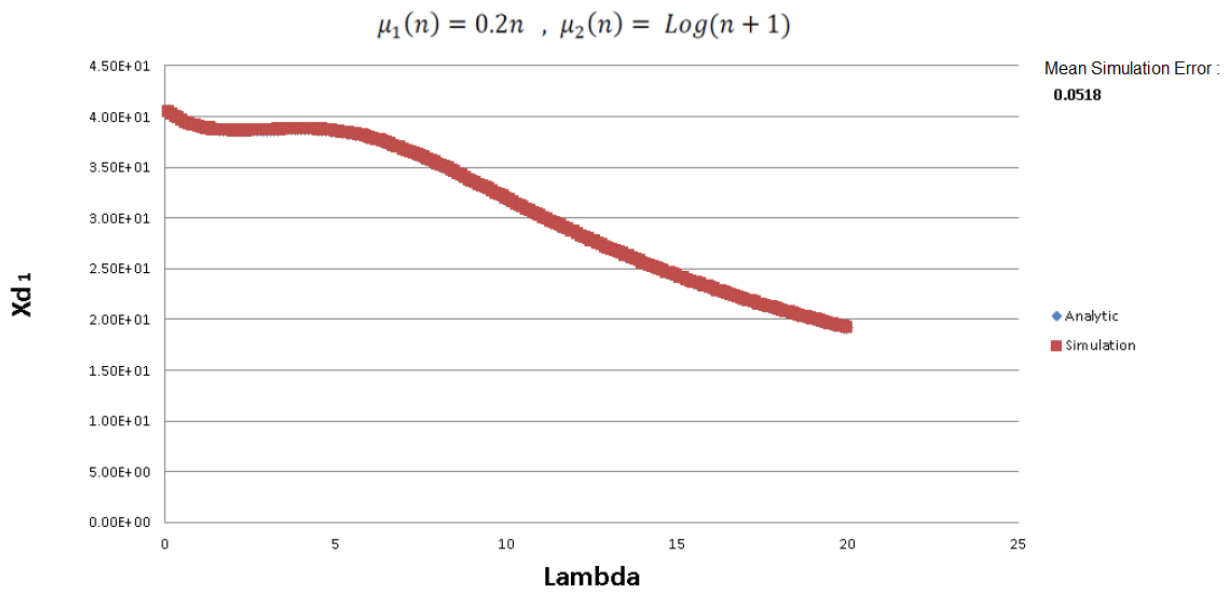
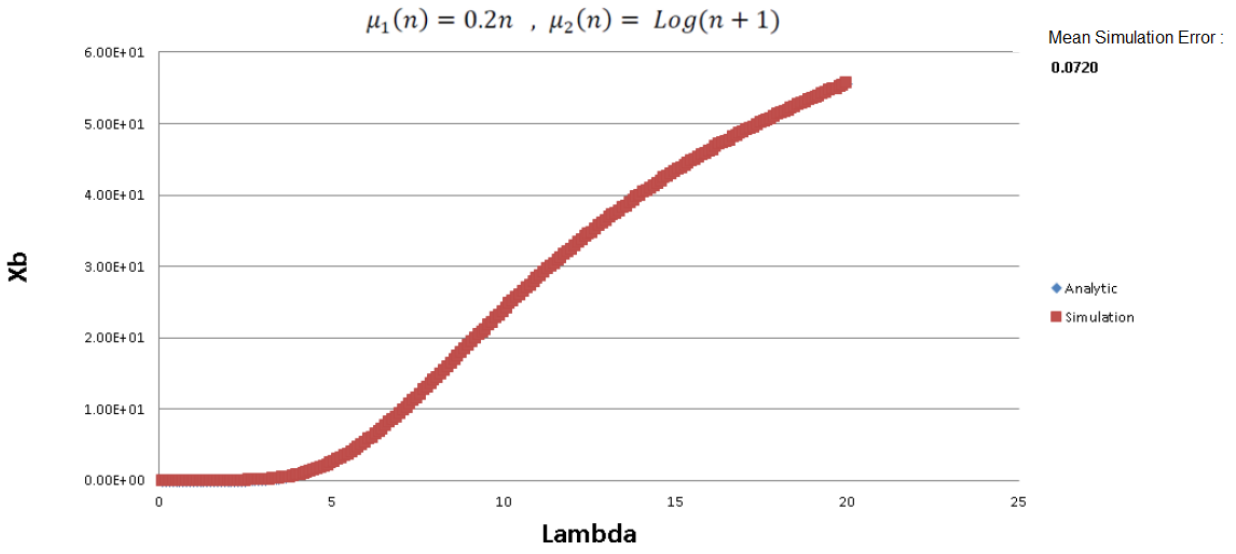


$$\mu_1(n) = \frac{\sqrt{n}+n}{10}, \mu_2(n) = \frac{\sqrt{n}+n}{10}$$

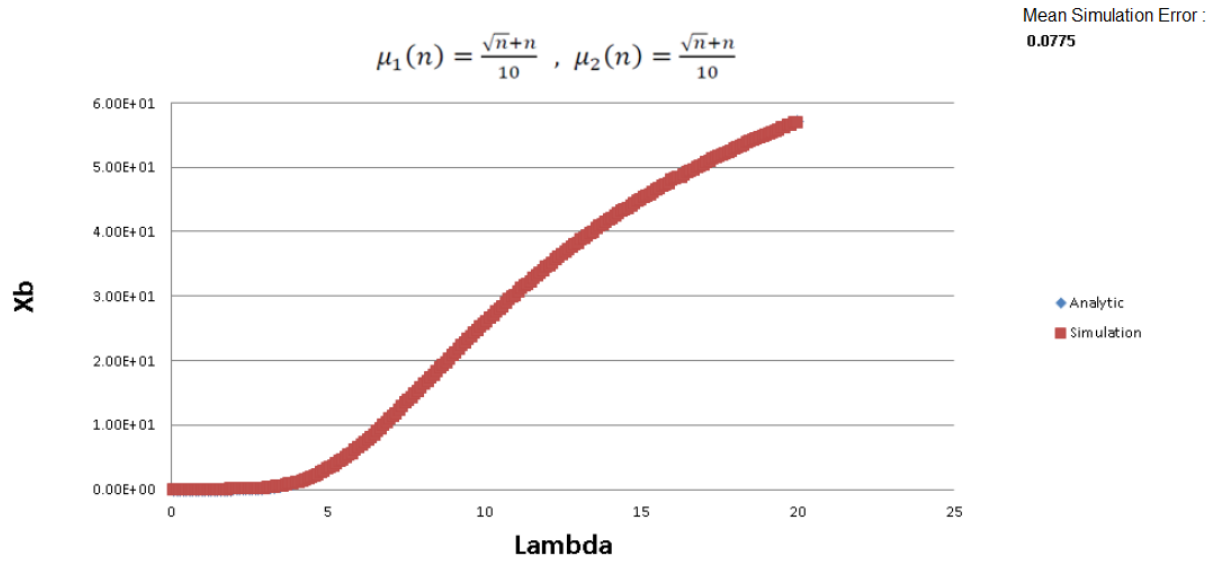
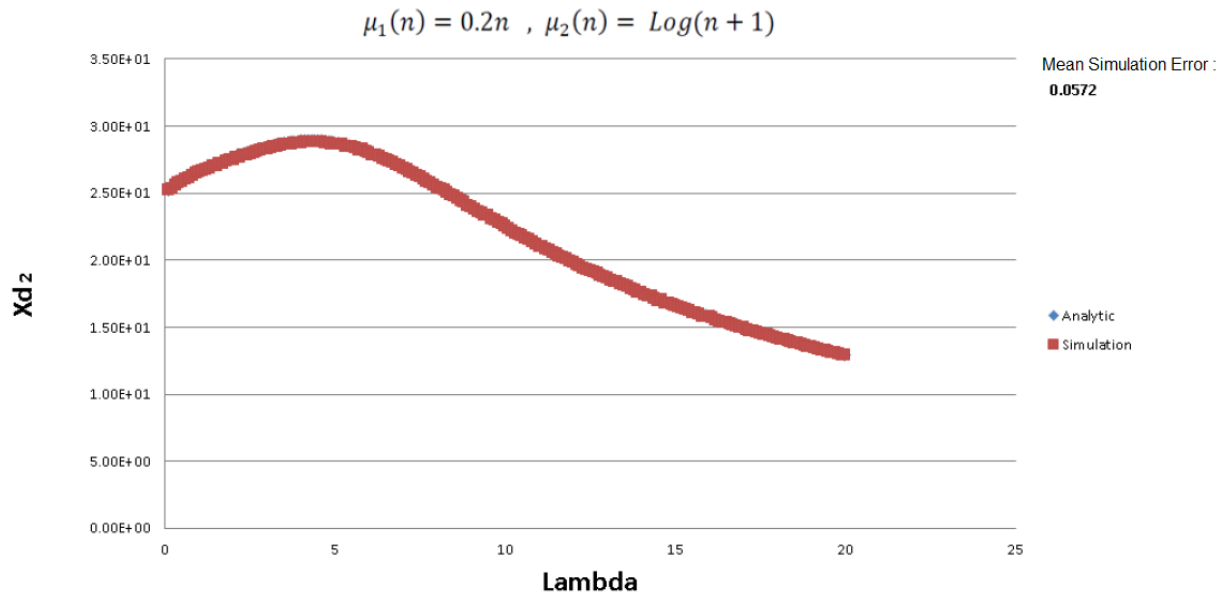


$$\mu_1(n) = \frac{\sqrt{n}+n}{10}, \mu_2(n) = \frac{\sqrt{n}+n}{10}$$



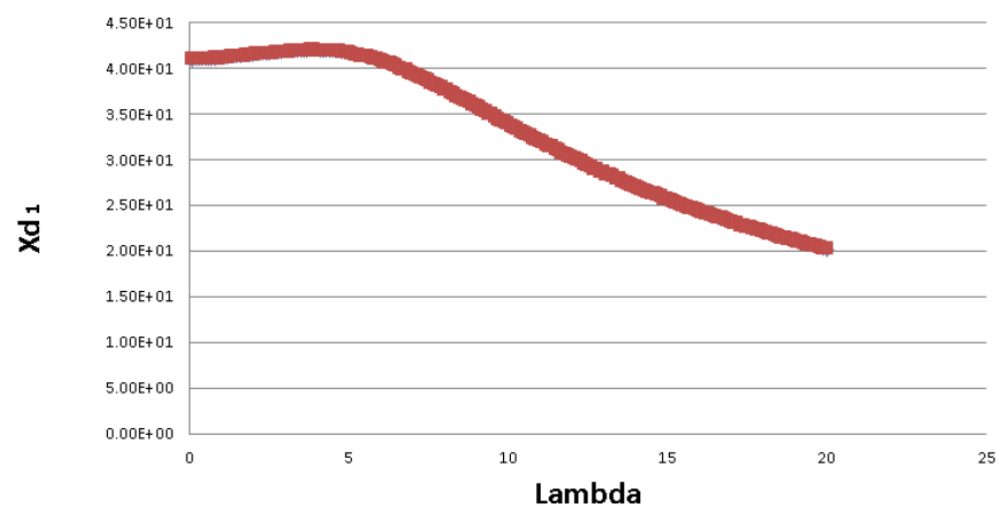






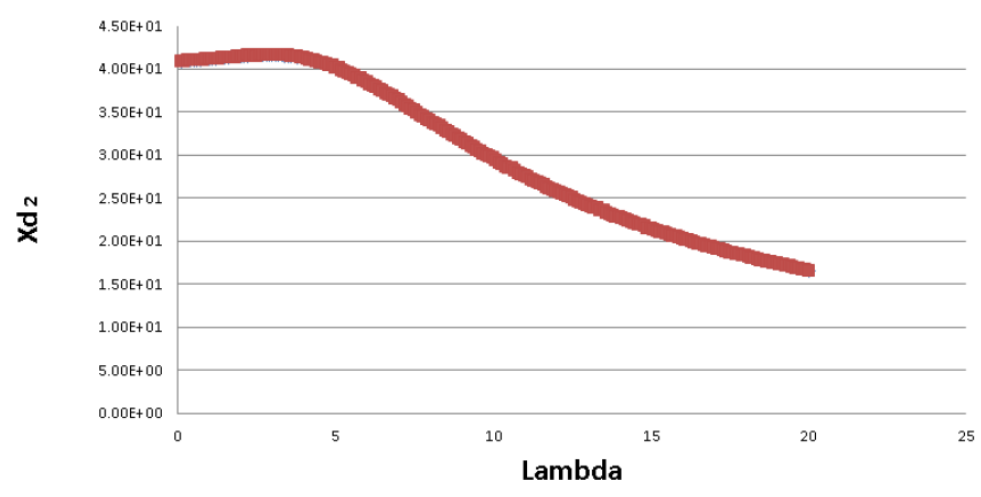
$$\mu_1(n) = \frac{\sqrt{n+n}}{10} , \mu_2(n) = \frac{\sqrt{n+n}}{10}$$

Mean Simulation Error :  
**0.0434**



$$\mu_1(n) = \frac{\sqrt{n+n}}{10} , \mu_2(n) = \frac{\sqrt{n+n}}{10}$$

Mean Simulation Error :  
**0.0334**



## **References**

- [1] Kant, Krishna, and K. Kant. *Introduction to computer system performance evaluation*. McGraw-Hill College, 1992.
- [2] Meyer, John F., Ali Movaghar, and William H. Sanders. "Stochastic Activity Networks: Structure, Behavior, and Application." In *Unknown Host Publication Title*, pp. 106-115. IEEE, 1985.
- [3] Banks, Jerry. *Discrete event system simulation*. Pearson Education India, 2005.
- [4] Nowroozi, Alireza, Peyman Teymoori, Toktam Ramezanifarkhani, Mohammad Reza Besharati, and Mohammad Izadi. "A Crisis Situations Decision-Making Systems Software Development Process With Rescue Experiences." *IEEE Access* 8 (2020): 59599-59617.
- [5] Movaghar, Ali, Lecture Notes, <http://ce.sharif.ir/courses/93-94/1/ce824-1/>
- [6] <https://www.mobius.illinois.edu/>
- [7] Daly, David, Daniel D. Deavours, Jay M. Doyle, Patrick G. Webster, and William H. Sanders. "Möbius: An extensible tool for performance and dependability modeling." In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pp. 332-336. Springer, Berlin, Heidelberg, 2000.