Concept of Simulation

Introduction

"Simulation is the imitation of operation of a real world process or system over time. Simulation can be done using a digital computer or by hand or any other equipment. In simulation we generate the artificial history of a system, and we observe the history to find out the different operational characteristics of the real system.

The behavior of the system according to time is studied by creating a system model. There are many assumptions to be made when we develop a model. The assumptions can be expressed in terms of mathematical, logical, symbolic or other relationships. After the development of the model it can be used to predict the changes in output by different changes in the parameters of the system, or it can be used in design stage to build any real world system. Thus "simulation is a tool that can be used to predict the effect of changes to existing systems, and as design tool to predict the performance of new systems under varying sets of circumstances."

Many times the model can be made in terms of mathematical expressions. Thus different mathematical techniques can be used to study the system. In these cases a numerical computer can be used to imitate the behavior of the system over time.

Data collected by the process of simulation can be used to estimate the measure of the performance of the system, as if it was collected from the real system.

System: A system is defined as group of objects that are joined together in some regular interaction or interdependence towards the accomplishment of some purpose. Which means that a system is group of objects (entities) that interacts with each other or has dependence on each other, and which perform different functions to achieve a common goal? For example in a production system the machines, component parts and workers work together to produce a high quality machine.

System boundary: A system boundary separates the system from the external environment. Thus all the entities that lie outside the system boundary are said to be in the environment. The system boundary is not a concrete term. It can change according to what we consider to be the system. For example in the case of the production line as above the change in price of the raw materials can be considered as a part of the environment. But if we want to study the effect of supply and demand then it must be considered to be the part of the system.

Entities: The different objects within the system that interact among one another are called system entities.

The property of entity is called an attribute of the entity. A specific length of time within the system is called an activity.

State: The state of a system is defined as the collection of variables that must be used to describe the system at any one time, in relation to the object of study. Event is the occurrence that may change the state of the system. If an event occurs inside the system then it is called endogenous and if it occurs outside the system then it is called exogenous.

Attributes: property of an entity is called its attribute.

Activity: is the time period of specified length.

Variable and state variables: a variable is a quantity whose value describes the value of some attribute of the system. The collections of all the variables that are necessary to describe the system completely are called the state variables.

Event: is defined as an instantaneous occurrence that may change the state of the system.

Endogenous and Exogenous: The activities and the events that occur inside the system are called endogenous and that occur outside the system are called exogenous.

Find some examples of system with figures of the models

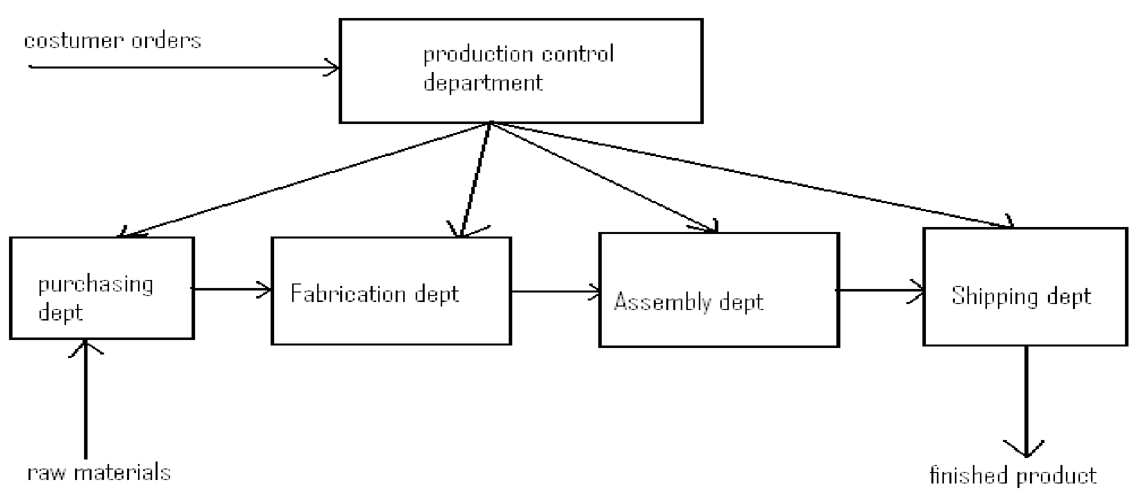


Fig: Factory system

The two major components of a factory system are fabrication department making the parts and the assembly department that produces the products. Purchasing department maintains a supply of raw materials and the shipping department receives orders and assigns work to the other departments. The entities are the departments, orders, parts, and products. The activities are the manufacturing processes of the departments. Attributes are such factors as the quantities for each order, type of part, or number of machines in a department.

System	***Entities***	***Attributes***	***activities***
Traffic	Cars	Speed Distance	Driving
Bank	Customers	Balance Credit status	Depositing
Communications	Messages	Length Priority	Transmitting
Supermarket	Customer	Shopping List	Checking-out

Continuous & Discrete Systems

Systems, in which the changes are predominantly smooth, are called continuous systems. For examples in an aircraft system with automatic pilot, d movement of the aircraft occurs smoothly. Thus this system can b considered as an example of the continuous system. We can also say that “a continuous system is one in which d state variables(s) change continuously over time.” Another example is d level of water behind a dam. During & for some time after a rain storm, water flows into d lake behind d dam. Water is drawn from the dam for flood control & to make electricity, this decreases d water level.

“A discrete system is the one in which d state variable(s) change only at a discrete set of points in time.”

Bank is an example of discrete system, here the state variable, the number of customers in the bank, changes only when a customer arrives or when d service provided to a customer is completed.

Few systems are wholly continuous or discrete. The aircraft for example may make discrete adjustments to its trim as altitude changes, & in a factory which can be considered as discrete, the machining proceeds continuously, even though the start & finish of a job are discrete. Complete aircraft can be considered as discrete, if the purpose of the study is to follow its progress along its scheduled route. In addition factory system can be considered to be continuously if number of parts in factory is large. When system itself is continuous but information about it is obtained only at some discrete points then it is called sampled data systems.

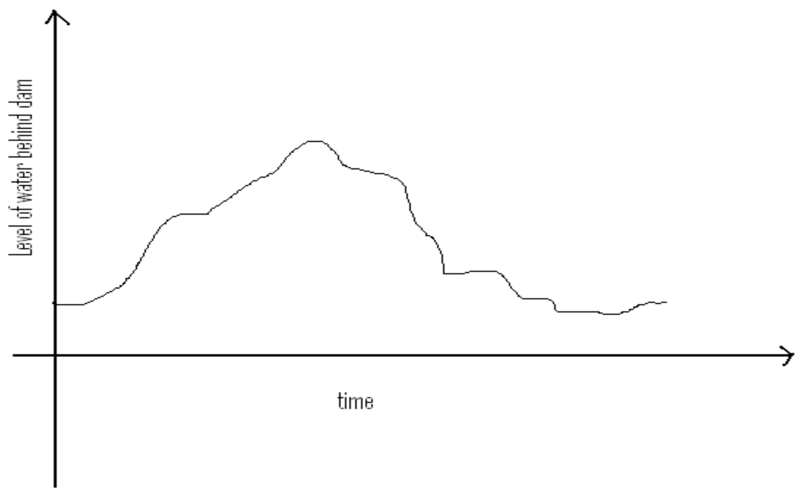


Fig: level of water behind dam, example of continuous system

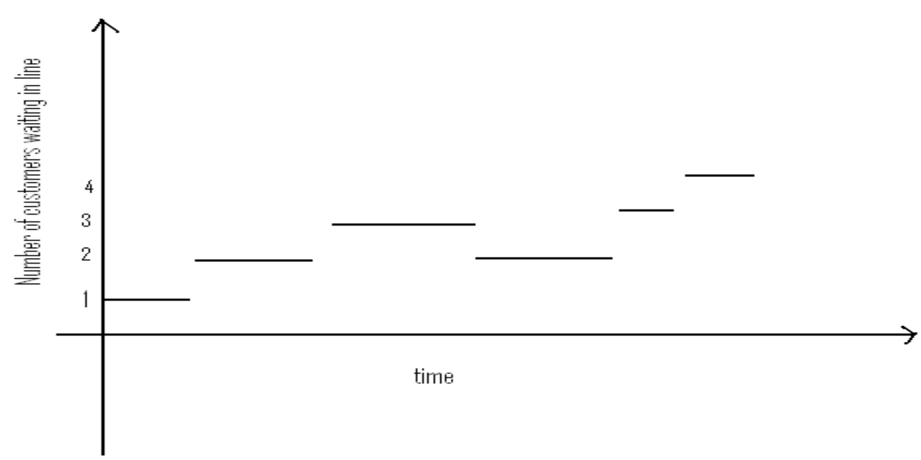


Fig: number of customers waiting in line in a bank, example of discrete system

When to use simulation

Simulation is used when

- We have to study a modern system which is so complex that the interactions can get rated only through simulation, such as the factory, service organizations etc.
- We want to train people for the jobs without disrupting the actual system and using less cost.
- We have to find the requirements of the machines by simulating their capabilities.
- We want to study and experiment with the interactions of a complex system, or of a subsystem within a complex system.
- We have to observe the effects of the informational, organizational, and environmental changes to the real system.
- When we want to introduce a new policy. Without the use of simulation we cannot know what effect it will have, thus by simulation we can find what possible outcome will occur after the policy is introduced.
- When we want to find out the output of a system without actually constructing it because many times the real system may be too costly to make and too time consuming.
- When we have to find the output of a long time in a very short time. For example to find out about the change of magnetic field of the earth in 10,0000 or more years we can use supercomputers which will produce results in a few months.
- When we want to study about the system which is very hard or impossible to investigate by actually doing experiment on it. For example if we have to find out about the effect of asteroid on other planets then it is very hard to go to other planets to study it, hence simulation is the best tool.
- When we want to find the effect of a very dangerous experiment or system without actually putting human life and environment at risk. For example the effect of explosion of a very powerful atomic bomb in an urban area cannot be done in reality as it puts too much human life at risk. Simulation should be used in such cases.

Advantages of simulation

1. New policies operating procedures, decision rules, information flows, organizational procedures, and so on can be explored without disrupting ongoing operations of the real system.

2. New hardware designs, physical layouts, transportation systems, and so on can be tested without committing resources for their acquisition.
3. Hypotheses about how or why certain phenomena occur can be tested for feasibility.
4. Time can be compressed or expanded allowing for a speedup or slowdown of the phenomena under investigation.
5. Insight can be obtained about the interaction of variables.
6. Insight can be obtained about the importance of the variables to the performance of the system.
7. Bottleneck analysis can be performed indicating where work-in-process, information, materials and son on are being excessively delayed.
8. A simulation study can help in understanding how the system operates rather than how individuals think it operates.
9. "What-if" questions can be answered useful in design of the system.

Disadvantages of simulation

1. Model building requires training. It is an art that is learned over time and through experience. Furthermore, if two models are constructed by two people they may have similarities by it is highly unlikely that both of them are exactly alike.
2. Simulation results may be difficult to interpret. Since most simulation outputs are essentially random variables (they are usually based on random inputs), it may be hard to determine whether an observation is a result of system interrelationships or randomness.
3. Simulation modeling and analysis can be time consuming and expensive.
4. Simulation is used in cases where analytical solutions are possible or even more important and preferable.

Areas of application

Areas of application for simulation are vast. The following shows the areas and the specific subjects within those areas that use simulation

- Manufacturing applications
  - Analysis of electronics assembly operations
  - Design and evaluation of selective assembly station for high-precision scroll compressor shells

    - Evaluation of cluster tool throughout for thin-film head production
    - Determining optimal lot size for a semiconductor back-end factory
    - Analysis of storage and retrieval strategies in warehouse
    - Model for army chemical ammunitions disposal facility.
- Semiconductor Manufacturing
    - Comparison of dispatching rules using large-facility models
    - The corrupting influence of variability
    - A new lot-release rule for wafer fabs
    - Comparison of 200-mm and 300-mm X-ray lithography cell
    - Capacity planning with time constraints between operations
- Construction Engineering
    - Construction of dam embankment
    - Activity scheduling in a dynamic, multiproject setting
    - Investigation of the structural steel erection process
    - Special-purpose template for utility tunnel construction
- Military Applications
    - Modeling leadership effects and recruit type in an army recruiting station
    - Design and test of an intelligent controller for autonomous underwater vehicles
    - Modeling military requirements for non war fighting operations
- Business Process Simulation
    - Impact of connection bank redesign on airport gate assignment
    - Product development program planning
    - Reconciliation of business and systems modeling
    - Personnel forecasting ad strategic workforce planning
- Human Systems
    - Modeling human performance in complex systems
    - Studying the human element in air traffic control

System Simulation: A mathematical model of a system is a model in which the relationship between the different entities and their attributes are expressed in the terms of mathematical expressions. When we are given a mathematical model of a system then we have two different methods that we can use to solve it. The first is the analytical method. This is the method we know from mathematics. Here different techniques such as the algebraically and geometric methods are used to find the general solution to the model. The analytical method gives the general solution which can be used to find the results of different situations.

The other method that we can use is the numerical method. In this method the solution is not general and the solution is got in steps. Each solution gives the solution of one condition, and the steps must be done several times if we want to find the solution to every condition.

The term “simulation” can be used to describe any procedure of establishing a model and deriving a solution numerically. However in the case of static models “simulation” is same as the “numerical computation”. In the case of dynamic models however we can differentiate between the “numerical method” and the “simulation”. The process of solving a dynamic system numerically by solving the equations of the model, step by step, with increasing value of time is called the “system simulation”.

For example the equation of the motion of a wheel can be given by

$$\mathbf{P^2 + 2\zeta\omega p + \omega^2 = 0}$$

By using the analytical method we can find that the oscillation of the wheel will not be present for $\zeta \geq 1$.

But instead of solving the problem analytically, if we observe the performance of the system for different conditions, then after doing a lot of experiments we will find that the oscillation of the wheel will not occur for all values of $\zeta \geq 1$. Hence after doing a lot of experiments we can find the solution for oscillation not to occur. This method is called the “system simulation”. It is a bit different from numerical analysis itself because in numerical analysis we do not care about the actual system itself. In the case of “system simulation” the real system performance is observed for different conditions and numerical analysis is used to find the performance.

The example of numerical computation method is the “Monte Carlo simulation method”.

Real time simulation:

While constructing the model of an engineering system we must understand the physical or chemical laws, and implies a number of experiments or measurements to derive the coefficients of the model. This can be very time-consuming if the model is not being simplified by assuming linearity. Thus the main difficulty arises because we must devote time and effort to understand the laws and derive the coefficients and most importantly we must make many assumptions about the system to make the model linear. This is a very time consuming preliminary work which can be avoided by using the real device instead of constructing the model. This type of approach is called the real time simulation.

In this technique actual devices, which are part of system under study, are used in conjunction with either a digital or hybrid computer. The computers provide the simulation for the parts that do not exist or that cannot be conveniently used in an experiment. Also in many simulation experiments we need to model the human being which is very hard, but in real time simulation it is achieved by interaction of a real human being. Along with the technology needed for hybrid simulation the real time simulation often use a real-time computer, thus we get the name real-time simulation. Real-time computers are the computers that can respond immediately to the signals sent from the physical device and send out signals at specific points in time.

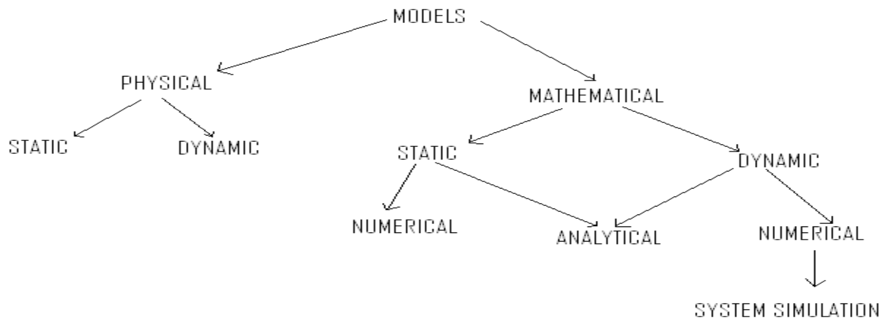
The real time simulation is used particularly in aerospace industry. These devices are called simulators and provide human beings with the alternative for some environment or situation. Example is a devices for training pilots by giving them the impression that they are at the controls of a real aircraft. Other well known example is the training of astronaut in ground for the zero-gravity condition.

Types of simulation model

A model is the body of information about a system gathered for the purpose of studying eh system. The purpose of the study will determine the nature of the information that is gathered, and there is no unique model of any system. For the same system the models made by different people may be different. It is also dependent on the nature of study.

To study a system it is sometimes possible to experiment with the system itself. The objective of many system studies, however, is to predict how a system will perform before it is built. It is not possible to experiment with a system when it is not in existence. We can build prototypes of the system and experiment with it, but it is very expensive and time consuming. For example, it is not practical to study economic system of a country by actually changing the supply and demands.

Models used in system studies have been classified in many ways. The figure of one such classification is shown in the figure



In this scheme the models are divided first into two types, physical and mathematical. Physical models are based on some analogy between system such as analogy between electrical a mechanical, electrical or hydraulic etc. In a physical model of a system, the system attributes are represented by such measurements as voltage or position of a shaft. The system activities are shown in terms of physical laws of another system. For example the rate at which the shaft a dc motor turns depends on the voltage applied. Thus if we have to model a car system then we can use the voltage applied as the velocity of the car then the number of revolutions of the shaft will give the distance traveled in given time.

Mathematical models use mathematical expressions and equations to represent the system. The system attributes are represented by variables and the activities are represented by mathematical functions that inter relate the variables.

The physical and the mathematical both are divided into two parts as static and dynamic. Static model only show the values that the system attributes take when system is in balance or equilibrium. The dynamic models show the value of the system as they are changing, or in transition period.

Static Physical model: As discussed above the static physical model uses analogy between systems to represent a system in terms of another and the system itself is in equilibrium. One well known example of the static physical model is the scale models. Scale models mean the miniature model of a large system such as a ship or large duplicate of a small thing such as the DNA molecule. For example to make a static model of a molecule we can use spheres to represent the atoms and rods or metal plates to represent the bond. This technique was used and had great success in deciphering the characteristics of the DNA molecule.

Sometimes the static physical model is used to solve the equation with one set of boundary condition. For example if we want measure the heat distribution in a complex shape, then we can make the shape and flow charge through it. Then we can measure the value of charges at different points. The value of charges gives the value of heat if we had flown heat instead of charge, because heat and charge distribute in analogous manner. Thus by studying the charge distribution we can find out about the heat property of the body.

Dynamic Physical model: in the static physical model the system is in equilibrium condition meaning that the values of the variables remain constant. In a dynamic model the state is constantly changing. Thus in dynamic physical model analogy between the system being studied and some other system is used, the analogy depends upon an underlying similarity in the forces governing the behavior of the system.

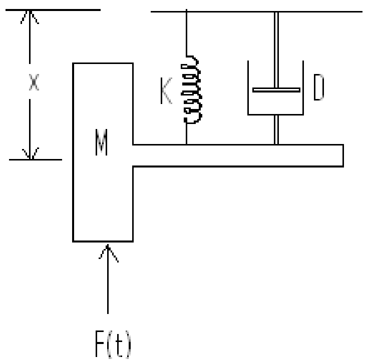
For example to study the mechanical system in which the force F is applied to the mass M with spring constant attached K and damping constant D and equivalent electric circuit can be created in which the voltage V(t) gives the force applied, inductance L is equal to M, resistance R is equal to D and 1/C where C is capacitance is equal to K. The value of charge q will give the distance moved by the mass, x.

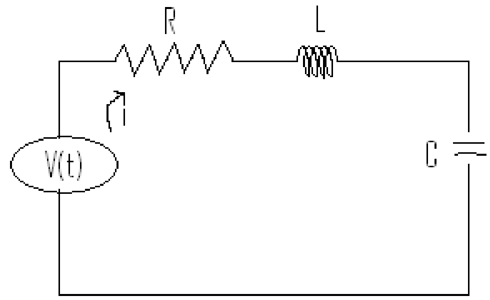
$$M\, d^2x/dt^2 + D\, dx/dt + Kx = KF(t)$$

Is analogous to

$$L\, d^2q/dt^2 + R\, dq/dt + q / C = V(t) / C$$

The mechanical and the electrical systems shown here are called the analogues and the performance of each can be studied with the study of other. In practice it is simpler to change the





electrical system to the mechanical system hence the mechanical system is usually studied with the help of the electrical system.

Static mathematical model: static mathematical model gives the values of the system variable when the system is in equilibrium in terms of mathematical equations and expressions. For example in the market there is a balance between demand supplies. Both of these factors depend on the price. A simple model can be made in which the balance between supply and demand will occur at the right price. Demand will be low when the price is high, and it will increase as the price drops. The relationship between the demand Q and the price P is called the demand curve. The supply increases as the price increases and hence as P increases supply S will increase. Thus the relationship between the supply and the price is called the supply curve. If we represent both the demand and the supply by straight lines then we can right

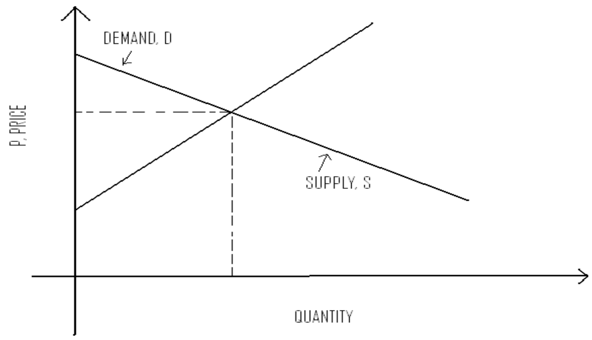
$Q = a – bP$

$S = c + dP$

$S = Q$ for equilibrium condition

If we put a = 600, b = 3,000 c = -100 and d = 2,000

Then $p = (a-c) / (b+d)$ hence $p = 0.14$

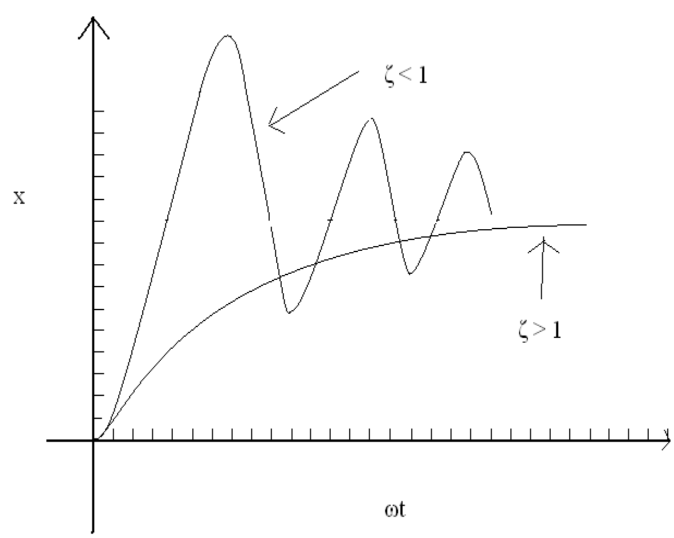


Dynamic Mathematical model: a dynamic mathematical model allows the changes of system attributes to be derived as a function of time. The equation that is used to describe the behavior of a car wheel is an example of a dynamic mathematical model. The equation is written as

$$\mathbf{d^2x / dt^2 + 2\zeta\omega\, dx/dt + \omega^2 x = \omega^2 F(t)}$$

Where $D/M = 2\zeta\omega$ and $\omega^2 = K / M$.

For ζ less than 1 the motion is oscillatory. This can be seen as how the x varies when we increase time t. Since it can be seen that the x varies with time t we know that the system is dynamic.

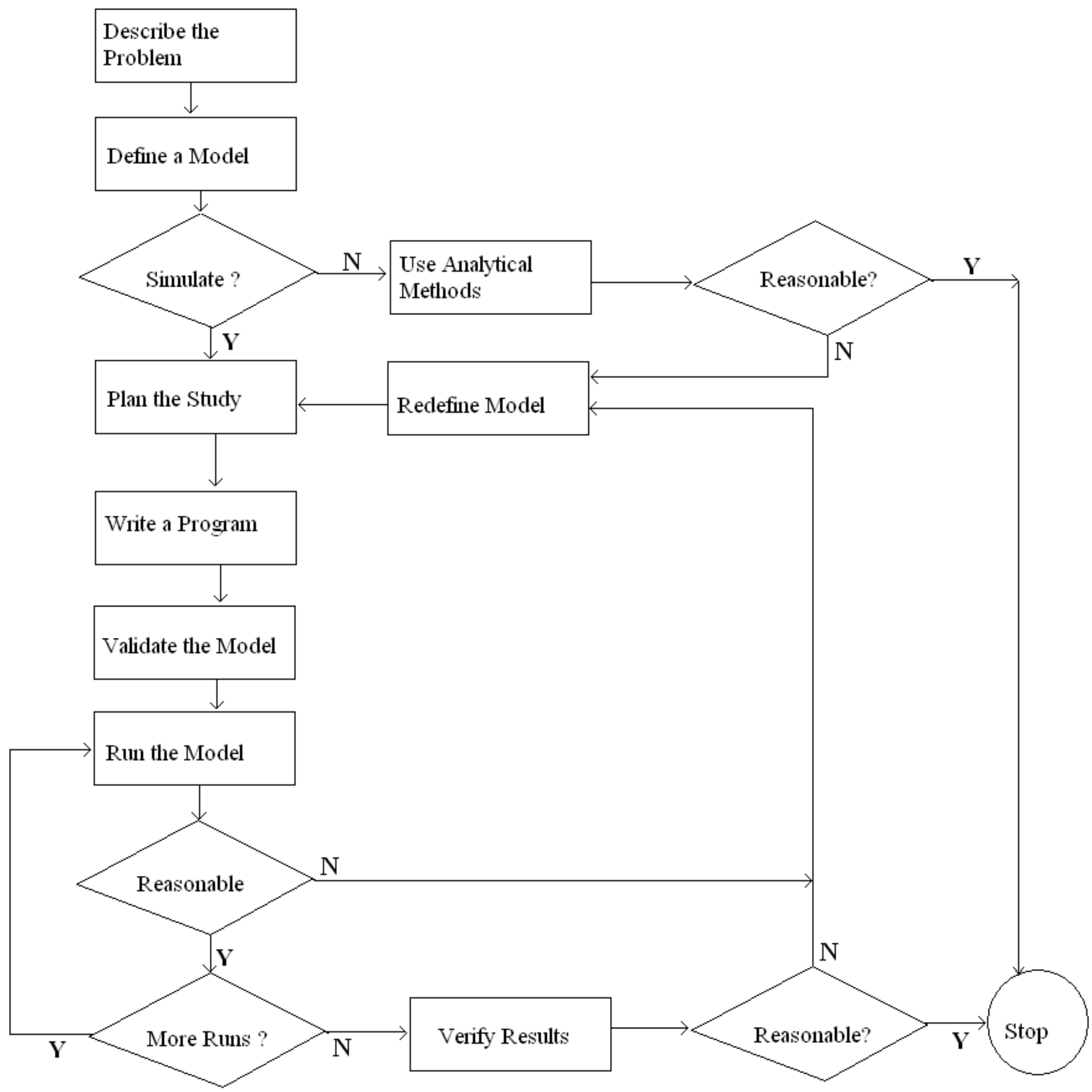


Steps in simulation study:

1) Problem Formulation
2) Model Construction

3) Plan the Study
4) Model Programming
5) Validation
6) Design of Experiment
7) Simulation Run n Analysis
8) Documentation
9) Implementation

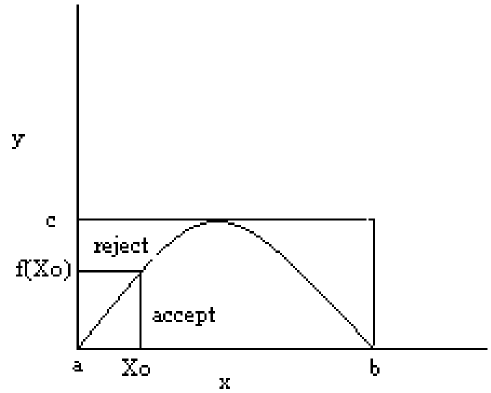
Initial step in the study of simulation is the problem formulation. In this step problem to be solved is described in a concise manner. The description of the problem of the system should be more enough to answer the questions asked and give the measurements need to be taken for the questions. Based on the problem definition, model of the system is created. In this stage we can decide a model can be kept in an analytic technique. It is possible that there is not a single model for a system. Many models may be used and should be constructed. Model constructed for the system guides for the simulation study i.e. this study helps to decide whether to do analytic method or to perform the simulation. Now, when it is decided to perform simulation, the nature of the simulation plans the study for the system. Plan the study decides the major parameter that is to be varied, the number of cases to be conducted and the order in which runs are to be made. The plan study may need to be revised periodically. Generally, simulation is performed in a digital computer, thus a program must be written. Then the validity of the program is checked for every step. However the validity should be checked for every step. After validation according to the plan and model of the system design for the experiment is done. The study will then move to executing series of runs according to plan. After simulation documentation of the study is performed so that it helps for the future implementation of the similar type of the problem. Documentation gives the actual results of the system which enables for its implementation i.e. the results from the simulation helps in the actual implementation of the system.



Flow Chart for the Simulation Process

Monte Carlo Method:

Monte Carlo method consists of experiment sampling with random numbers. The Monte Carlo technique is a computational technique applied to static models .e.g. the integral of a single variable over a given range corresponds to finding the area ender the graph representing the function.



Here, f(x) is the positive function bounded with lower and upper bounds a and b respectively. The function is bounded above by c. as shown in figure the function is contained within a rectangle with length c and (b-a). We pick the random points within the rectangle and determine whether they lie beneath the curve or not. The fraction of points falling on or below the curve to the area of rectangle. i.e.

$$\frac{n}{N} = \frac{\int_a^b f(x)dx}{c(b-a)}$$

Here, n is the number of points that lie on or below the curve. N is the number of generated points.

As the number N increases the accuracy of the results increases. For sufficient points the value of the integral is estimated by multiplying n/N by the area of rectangle.

$$\int_a^b f(x)dx = \frac{n}{N} * c(b-a)$$

For each point, a value of x is selected at random between a & b, say Xo. A second random number is between 0 to c to give y. if y≤ f(Xo) the point is accepted in n else rejected.

e.g.

Solve $\int_2^4 x^2 dx$ using Monte Carlo method and also find the error percentage.

Here, $y=f(x)=x^2$

Lower limit=2

Upper limit =4

X	2	3	4
y	4	9	16

Area of Rectangle= (4-2)*16

=32

Suppose, we take N=5 random dots (x, y) inside the rectangle such that

$2 \leq x \leq 4$

$0 \leq y \leq 16$

$(x_1, y_1)=(2, 0)$

$(x_2, y_2)=(2, 2)$

$(x_3, y_3)=(4, 5)$

$(x_4, y_4)=(3, 10)$

$(x_5, y_5)=(2, 15)$

Now, we can write the equation of curve as

$y \leq x^2$

Putting the value of x and y in above equation,

For (2, 0); $0 \leq 4$, so (2, 0) is accepted.

For (2, 2); $2 \leq 4$, so (2, 2) is accepted.

For (4, 5); $5 \le 16$, so (4, 5) is accepted.

For (3, 10); $10 \le 9$, since it is false, (3, 10) is rejected.

For (2, 15); $15 \le 4$, since it is false, (2, 15) is rejected.

Since the accepted value lies inside the curve and the rejected value lies outside the curve but inside the rectangle N=5 and n=3

Now, using Monte Carlo Method

$$\frac{\text{Area under curve}}{\text{Area under Rectangle}} = \frac{n}{N}$$

So, Area under Curve = 3/5*32

= 19.2

Comparison of Simulation and Analytic method

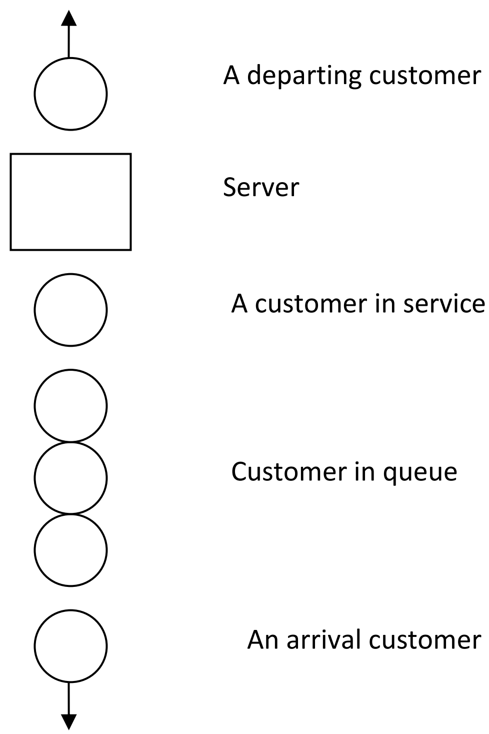
1) Simulation gives specific solution rather than a general solution. This is the advantages of analytical method over simulation.
2) Simulation evolves numerical method where as problem are solved in analytic method by deductive reasoning.
3) The range of problem that can be solved by analytic method is limited. Simulation can solve almost any problem.
4) For the real world problems simulation is more preferred than analytic method.
5) For more complex problem simulation is faster than analytic method.
6) Accuracy rate is more in analytic method than simulation.
7) For complex problem analytic method is hard to apply.

Stochastic simulation

The simulation system where the effects of activity varies randomly to produce various possible output, it is stochastic simulation. The random output of stochastic simulation can be measured and described in the form of probability distribution. The occurrence of a random activity can be the part of the system. For e.g. in a factory system the time for machine operation can be defined as probability distribution but machining can be defined ad endogenous activity. There may be power failure in the random interval of time, and then it is exogenous activity.

Single Server Queuing Model:

Let us consider a single server queuing system for which inter arrival times A1, A2…… are independent and identically distributed random variables. A customer who arrives and finds server idle enters service immediately, and the service time S1, S2….of the successive customer are IID random variables. A customer who arrives and finds server busy joins the end of a single queue. When service for a customer is finished, the server chooses a customer form the queue in a FIFO manner.



Problem statement:

A Single Server Queuing System

Assumption:

1) Simulation begins in empty and idle state (no customer present). Server is idle.
2) At time 0, we will begin waiting for the customer, which will occur after the first inter arrival time A1 rather than at time 0.

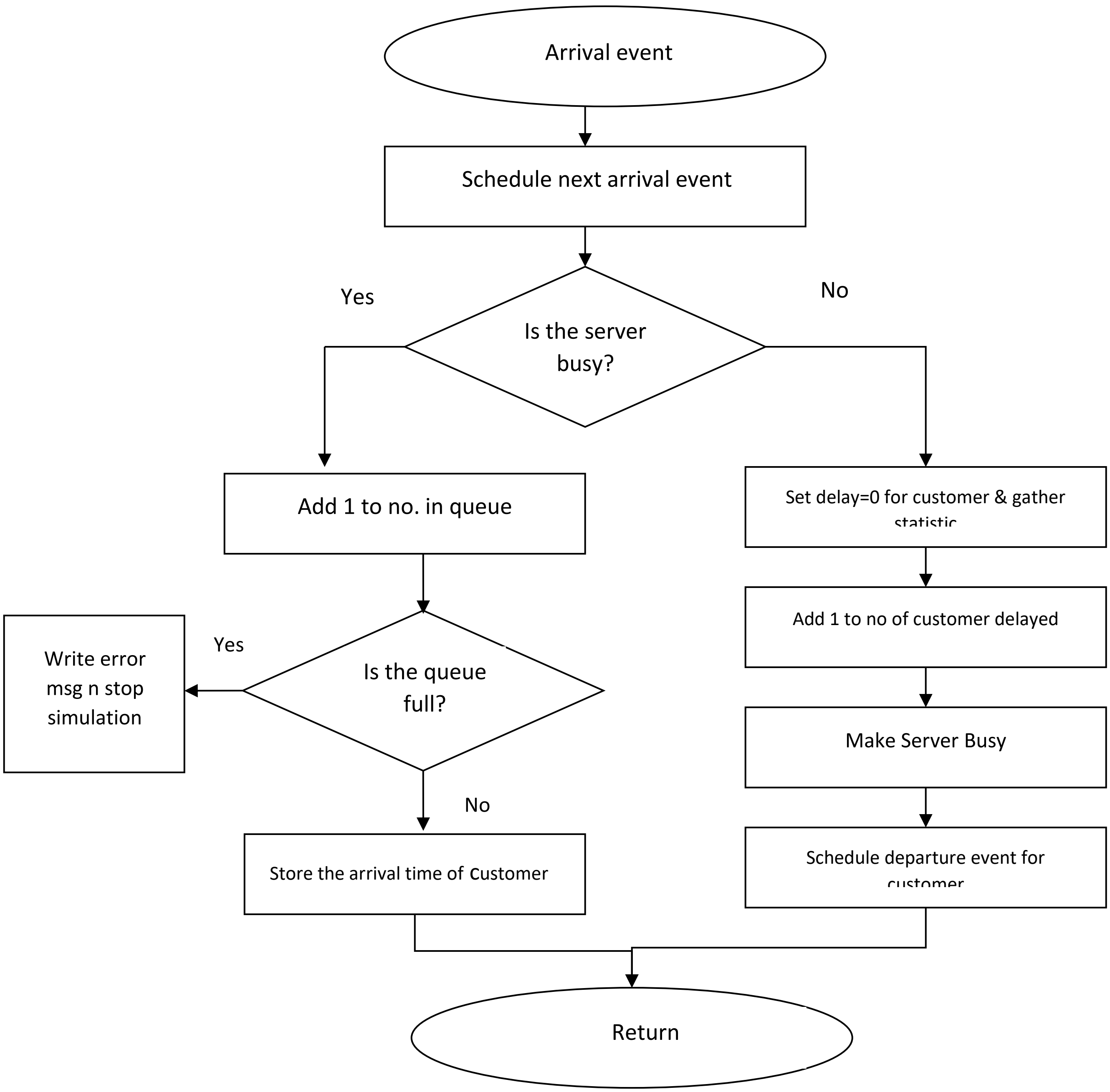
3) Simulation is done until a fixed number (n) of customers have completed their delays in queue, i.e. simulation stops when nth customer enters service.

Program Structure:

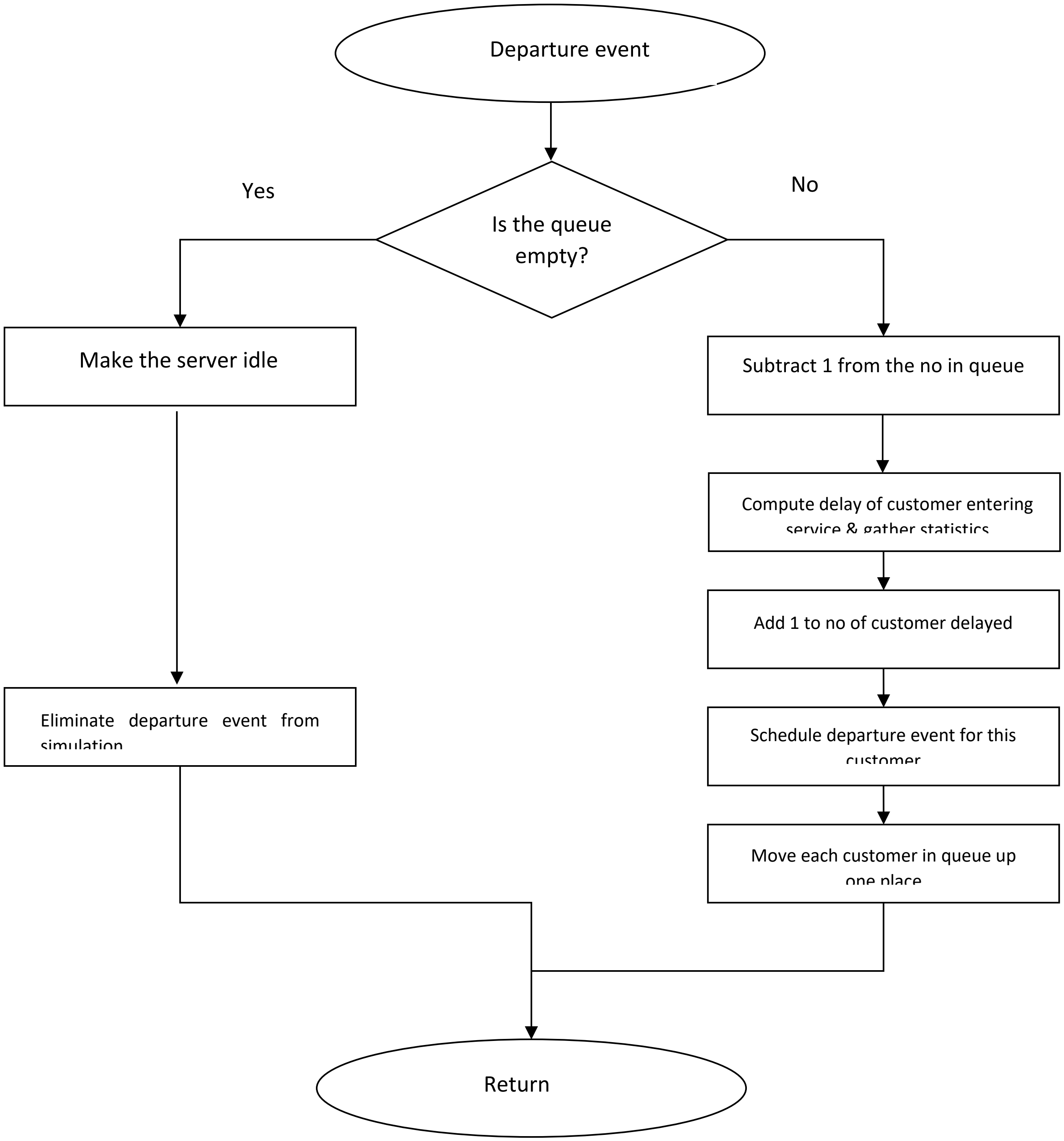
Mainly program contains two routines:

1) Arrival
2) Departure

Flow chart for arrival routine:



Flow chart for departure:



Performance measure of queuing system:

To measure the performance of the system we estimate three quantities:

1) Expected average delay in queue of the n customers completing their delays during the simulation. It is denoted by d (n).d (n) depends upon the inter arrival and service time random variables observations on a given run. For another run, arrival time may be different and different service time. Thus there will be different delay in n. Average delay on a given run of simulation is random variable itself.

   D1, D2, D3…………Dn is customer delays.

$$\hat{d}(n) = \frac{\sum_{i=1}^{n} D_i}{n}$$

Where, Di is the delay of customer i in the queue

For first customer when the server is empty and idle D1=0 as the queue is also empty initially.

$D_i \geq 0$

d(n) gives the system performance from the customer from the customer point of view.

2) Time-average number of customer in queue,

   Let Q (t) denotes the no. of customers in queue at time t, for any real number t≥0 and let T (n) be the time required to observe n delays in queue. Then Q (t) is a non negative integer for any time between 0 & T (n).

$$q(n) = \sum_{i=o}^{\infty} i\, p_i \qquad \text{……….(1)}$$

where pi is expected proportion of time & Q(t)= i

Let Ti be the total time taken during the simulation that the queue is of length i, then T(n)= T0+T1 +……. and pi=Ti/T(n).

$$q(n) = \frac{\sum_{i=o}^{\infty} i\,Ti}{T(n)} \qquad \ldots\ldots\ldots(2)$$

As we know simulation can be represented as integration so,

$\sum_{i=o}^{\infty} i\,Ti$ will be written as

$$\sum_{i=o}^{\infty} i\,Ti = \sum_{t=0}^{T(n)} Q(t)\,dt \qquad \ldots\ldots\ldots(3)$$

Thus,

$$q(n) = \sum_{t=0}^{T(n)} Q(t)\,dt \qquad \text{from equation (3) and (4)}$$

$$= \int_{0}^{T(n)} \frac{Q(t)\,dt}{T(n)}$$

So, q(n) is area under the curve Q(t) between the beginning and end of simulation.

3) Expected utilization of server[u(n)]

The expected utilization of the server is the expected proportion of time during the simulation i.e. from time 0 to T (n) for server's busy and is thus a number between 0 & 1.

Let B (t) denote the state of the server at time t.

$$B(t) = \begin{cases} 1 & \text{if the server is busy at time t.} \\ 0 & \text{if the server is idle at time t.} \end{cases}$$

Ti denotes the ith time interval for which the server is busy. Thus

$$u(n) = \frac{\sum_{i=0}^{\infty} T_i}{T(n)} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(a)$$

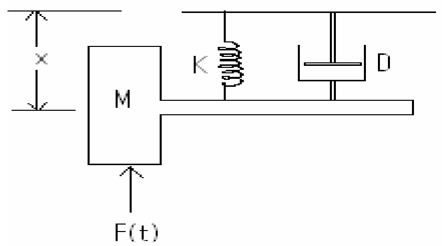
In integral form

$$u(n) = \frac{\sum_{i=0}^{T(n)} B(t)\,dt}{T(n)}$$

$$= \frac{\int_{0}^{T(n)} B(t)\,dt}{T(n)} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(b)$$

Differential Equations for Mechanical System:

This equation gives the differential equation for mechanical system describing the suspension wheel of an automobile.



Here x- displacement

M-mass

K-stiffness of spring

D-damping force

Here, for vertical displacement of wheel, x is the displacement of point, taking x to be positive for an upward movement.

The velocity of wheel in the vertical direction is rate of change of position which is first derivative of x. the acceleration of the wheel in vertical is rate of change of velocity is second derivative of x.

The system determines the relationship between applied force and movement of body i.e. acceleration of body is directly proportional to force.

$kF(t)=M\, d^2x/dt^2$ where M is the coefficient of proportionality .

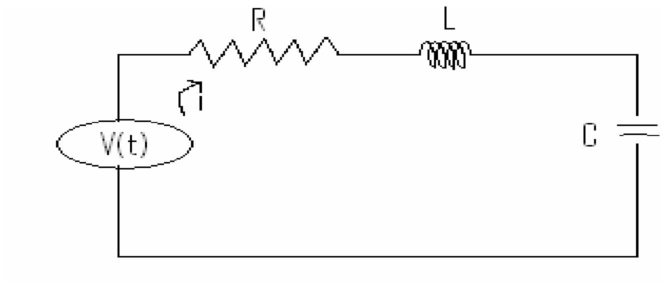
Shock absorber exerts a resisting force that depends on the velocity of the wheel i.e. force is at zero when wheel is at rest, and it increases as velocity increases.

For force directly proportional to velocity, it can be represented by D dx/dt , where D is measure of viscosity of shock absorber. Spring also exerts resisting force. So, force is also proportional to compression of spring i.e. kx where k is stiffness of spring constant.

$M\ d^2x/dt^2 = KF(t) - D\ dx/dt - Kx$

Differential Equations for Electrical System:

This equation gives the differential equation for electrical system comprising of resistor, inductor and capacitor.



In the above circuit,

R- Resistance in the circuit

L- Inductance in the circuit

V(t)- Voltage source function varying with time

C- Capacitance of a capacitor

i- Current flowing through the circuit

q- Charge

In above circuit current is the rate of change in charge i.e.

$i = dq/dt$ -------------------(1)

Let V_R, V_L, V_C be the voltage drop across resistor, inductor and capacitor. In RLC circuit current across resistor is directly proportional to V_R.

i.e. $i \infty V_R$

$V_{R=}$ R*i where R is the resistance of resisitor.

V(t) is the voltage source and di/dt is the rate of change of current in the circuit.

So voltage drop across inductor is L di/dt

i.e. V_L =L di/dt where L is the inductance

Voltage drop across capacitor is directly proportional to charge.

i.e. $V_C = q/C$;

Now, from KVL for above circuit

$V(t)= V_R+ V_L + V_C$ ------------------(2)

From equation 1 and 2

$V(t)= R\, dq/dt + L\, d^2 q/dt^2 + q/C$

This is the required equation for RLC circuit which is equivalent to the mechanical system.

Discrete Event Simulation

Discrete event simulation concerns the modeling of a system as it evolves over time by a representation in which stat variables change instantaneously at separate points in time. Here event is defined as an instantaneous occurrence that may change the state of the system. Discrete event simulation can be done by hand calculation. But to solve real world systems data must be stored to simulate it on a digital computer.

Eg:- A service facility with a single server. A one operator barbershop or an information desk at an airport. Here we would like to estimate the average delay of arriving customers where the delay in queue of a customer is the length of time interval from the instant of his arrival at the facility to the instant he begins served. To calculate the average delay of a customer, the state variables in a discrete- event simulation depends upon the condition of server i.e. either the server is busy or idle, the number of customers waiting in queue to be served or the arrival time of customer. Arrival time of customer is needed to compute is delay. This time is given by the service time minus arrival time. Here in this system two types of event occurs:

i) The arrival of customer
ii) Completion of service of customer

Arrival is an even as it changes the status of server from idle to busy and increases the number of customer in queue.

Similarly departure is an event since it changes the state of server from busy to idle and decreases the number of customer in queue

Components and organization of a Discrete-Event Simulation Model

1) System State- The collection of state variables necessary to describe the system at a particular time.
2) Simulation Clock- A variable giving the current value of simulated time.
3) Event list- A list containing the next time when each type of event will occur.
4) Statistical Counters- Variables used for storing statistical information about system performance.
5) Initialization Routine- A subprogram that initialize the simulation model at time 0.
6) Timing Routine- A subprogram that determines the next event from the event list and then advances the simulation clock to the time when the event is to occur.
7) Event Routine- A subprogram that updates the system state when a particular type of event occurs.
8) Library Routines- A set of subprograms used to generate random observations from probability distributions that were determined as part of simulation model.
9) Report Generator- A sub program that computer estimates of the desired measures of performance and produces a report when the simulation ends.
10) Main Program- A subprogram that invokes the timing routine to determine the next event and then transfers control to the corresponding event routine to update the system state appropriately. It also checks the termination and invokes the report generator when the simulation is over.

Generation of Arrival Patterns

Arrival pattern for particular system is specified for simulation. The exogenous arrivals can be designed for simulation.

Trace Driven Simulation

The sequence of inputs can be generated from the observation on a particular system are tested from record gathered from a running system, i.e. representative of the sequence of operations the computer system will have to execute. This method is trace driven simulation. Here program monitors can be attached to the running system to extract data with no or little disturbance to running system.

Bootstrapping Method

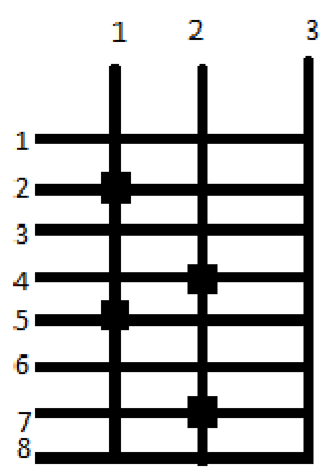
The arrival time of an entity is recorded as one of the event times. When simulation clock time reaches this event time, the event entering in the system is executed and the arrival time of next entity is calculated. This method is called bootstrapping method. Here one entity create its successor.

Telephone call Simulation as lost call System

This system represents telephone system where call gets lost when call cannot be connected at the time of its arrival. Call gets lost in two condition. When line is not available i.e. destination call is busy call gets lost and the other condition is when link is not available. Link is the connection between two lines.

Let us make some assumption to simulate the system

1. Let n= no of total link then line=2*n+2
2. Arrival process is bootstrapping method (simulation clock is updated at every arrival that may be either call connect or call disconnect)



Let us show the various state of simulation for following telephone calls. Here recently 2 is calling 5 and 4 and 7 is calling. In the various system state lines shows the status of lines. Lines those are busy are indicated by 1 and the line those are free are indicated by 0. Links indicates the maximum number of link and link in use. Call in progress indicates the status of ongoing call. Processed, completed, block and busy gives the counter of calls. Next arrival indicates the arrival of next call.

Lines
1-0
2-1
3-0
4-1
5-1
6-0
7-1
8-0

Clock	12

Next arrival		
From	To	Length
1	7	20
Arrival time-20		

Call in progress		
From	To	End
4	7	35
2	5	18

Links	
Max	3
In use	2

Processed	Completed	Block	Busy
2	0	0	0

System state-1

In system state 2 the, when the simulation clock reaches time 18, call 2-5 is completed and thus they are removed from call in progress, their state is change to 0 from 1, link in use is decremented by 1 and counter of call completed is increased by 1.

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	18

Call in Progress		
From	To	End
4	7	35

Next arrival		
From	To	Length
1	7	20
Arrival time-20		

Processed	Completed	Block	Busy
2	1	0	0

System state-2

In the system state 3, when simulation clock reached 20, line 1 tries to call line 7 which is still in call in progress and the connection is abandoned. In this case call counter of busy is incremented by 1 and all other remains same. Here call gets lost since line is busy.

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	20

Call in progress		
From	To	End
4	7	35

Next arrival		
From	To	Length
3	6	15
Arrival time-25		

Processed	Completed	Block	Busy
3	1	0	1

System state-3

Different states goes in the similar manner. If the line is busy call gets lost else till the link is available call connects. When call in progress is full that is max link = link in use, then call gets block as in the case of system state 6.

Lines
1-0
2-0
3-1
4-1
5-0
6-1
7-1
8-0

Links	
Max	3
In use	2

Clock	25

Call in progress		
From	To	End
3	6	40
4	7	35

Next Arrival		
From 2	To 8	Length 15
Arrival time-28		

Processed	Completed	Block	Busy
4	1	0	1

System state-4

Lines
1-0
2-1
3-1
4-1
5-0
6-1
7-1
8-1

Links	
Max	3
In use	3

Clock	28

Call in progress		
From	To	End
2	8	43
3	6	40
4	7	35

Next Arrival		
From 1	To 5	Length 15
Arrival time-30		

Processed	Completed	Block	Busy
5	1	0	1

System state-5

In this state since all the links are full, call cannot be connected though the lines are free. In this case also call gets lost and the call counter of block is increased by 1.

Lines
1-0
2-1
3-1
4-1
5-0
6-1
7-1
8-1

Links	
Max	3
In use	3

Clock	30

Call in progress		
From	To	End
2	8	43
3	6	40
4	7	35

Next Arrival		
From	To	Length
1	5	15
Arrival time-45		

Processed	Completed	Block	Busy
6	1	1	1

System state-6

Delayed call system

Let us modify the telephone call system so that the call that cannot be connected at the time of arrival does not get lost. They wait until they cannot be connected. The system is like a message passing switching system with store and forward capability.

To record of delay it is necessary to build another list like call in progress. For this we create another list, delay call list as shown in the system state 1. The first two states are same as lost call system.

Lines
1-0
2-1
3-0
4-1
5-5
6-0
7-1
8-0

Links	
Max	3
In use	2

Clock	12

Next arrival		
From	To	Length
1	7	15
Arrival time-20		

Call in progress		
From	To	End
4	7	25
2	5	15

Delay Call list		
From	To	Length

Processed	Completed	Block	Busy
2	0	0	0

System state-1

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	15

Call in progress		
From	To	End
4	7	25

Delay Call list		
From	To	Length

Next arrival		
From	To	Length
1	7	15
Arrival time-20		

Processed	Completed	Block	Busy
2	1	0	0

System state-2

Now, when call is completed it is necessary to check the list call for waiting call. If there is any call the call is forwarded to call in progress. If there is not any next arrival call is processed. At simulation clock 20, 1 calls 7 which is busy till clock time 25. Thus it is stored in delay call list at shown in system state 3.

Lines
1-0
2-0
3-0
4-1
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	20

Next arrival		
From 8	To 6	Length 20
Arrival time-30		

Call in progress		
From	To	End
4	7	25

Delay Call list		
From	To	Length
1	7	15

Processed	Completed	Block	Busy
3	1	0	1

System state-3

The system goes to next state with record of delayed call. At time 25 call from 4 to 7 is completed. Now, 7 is free. Thus call from delayed list is connected which is from 1 to 7. Now, the record are updated. Delayed call is transferred to call in progress list and the system state change like this.

Lines
1-1
2-0
3-0
4-0
5-0
6-0
7-1
8-0

Links	
Max	3
In use	1

Clock	25

Next Arrival		
From 8	To 6	Length 20
Arrival time-30		

Call in Progress		
1	7	40

Delay Call list		

Processed	Completed	Block	Busy
3	2	0	1

System state-4

Similarly block condition can be shown and call connection can be done.

Simulation programming task

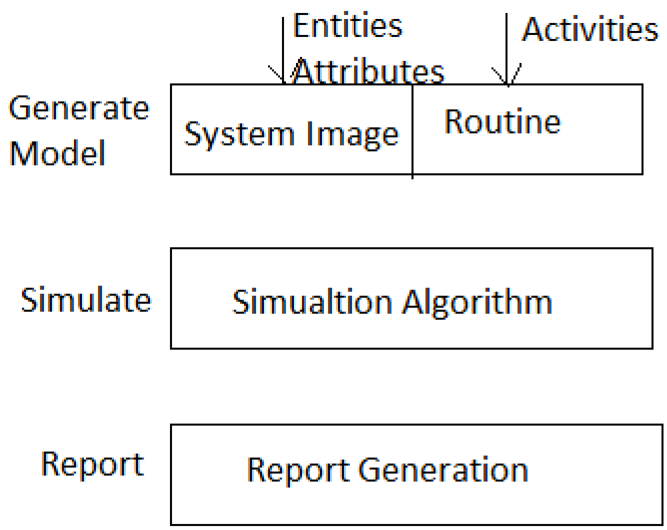
There are three main task to be performed.

i) Generate Model

ii) Simulate

iii) Report

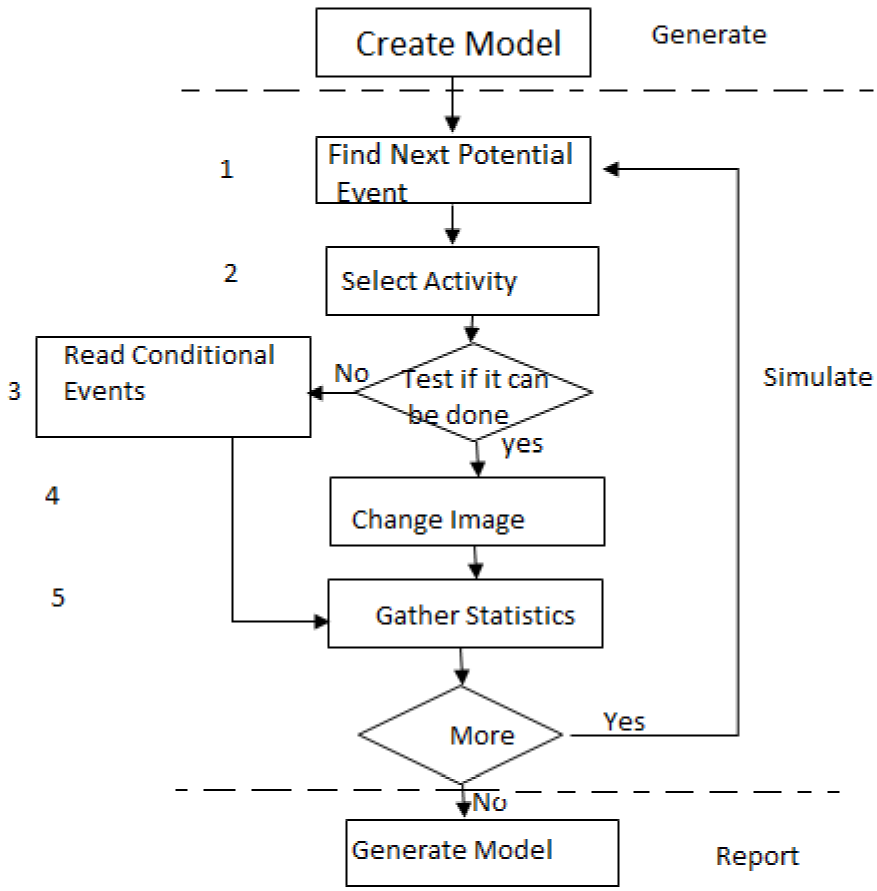
The first is to generate a model and initialize it. From the description we generate system image which is the set of numbers that elects the state of the system all times. The second task is to program the procedure that executes cycle of actions to carry out simulation. This process is simulation algorithm. The third task is report generation.



Simulation Programming Task

To carry out simulation algorithm steps are:

1. Find the next potential event
2. Select an activity
3. Test if the event can be executed
4. Change the system image
5. Gather statistics



Flow chart

Gathering Statistics

Commonly required statistics are:-

1. Counts – gives the number of entities or number of times event occurred.
2. Summary Measures- gives extreme values, mean values and standard deviations.
3. Utilization- time some entity is engaged.
4. Occupancy- groups of entities used on average.
5. Distributions- variables such as queue length or waiting times.
6. Transit times- defines as time taken for an entity to move from one part of the system to other part

Counters and Summary measures-

Counters are the basis for most statistics. Some accumulates total and some record the current values in the system. The telephone system simulation use counters to record the total number of lost and busy calls. Maxima and minima are easily obtained.

The mean of a set of N observations X_r = 1, 2,3,……, N

$$m = 1/N \sum_{r=1}^{n} X_r$$

Standard deviation is defined as

$$s = \sqrt{\frac{1}{N-1}\sum_{r=1}^{N}(m - x_r)^2}$$

Common method is

$$\sum_{r=1}^{N}(m - x_r)^2 = \sum_{r=1}^{N} x_r^{\,2} - Nm^2$$

Measuring utilization and occupancy

Utilization is used to describe what fraction of time the item is engaged during simulation run. To measure utilization, record of the time t_b is necessary which gives the time at which the item last become busy. When the entity becomes free at time t_f the interval $t_f - t_b$ is added to a counter. At the end of simulation run, the utilization U is derived by dividing the accumulated total time T for entity used N times.

$$U = 1/T \sum_{r=1}^{N}(t_f - t_b)_r$$

For discrete system $t_f - t_b$ is measure directly. For continuous system count is given by counting the number of intervals in which the item is busy. Fro group of entities, it also requires the information about numbers of entities involved.

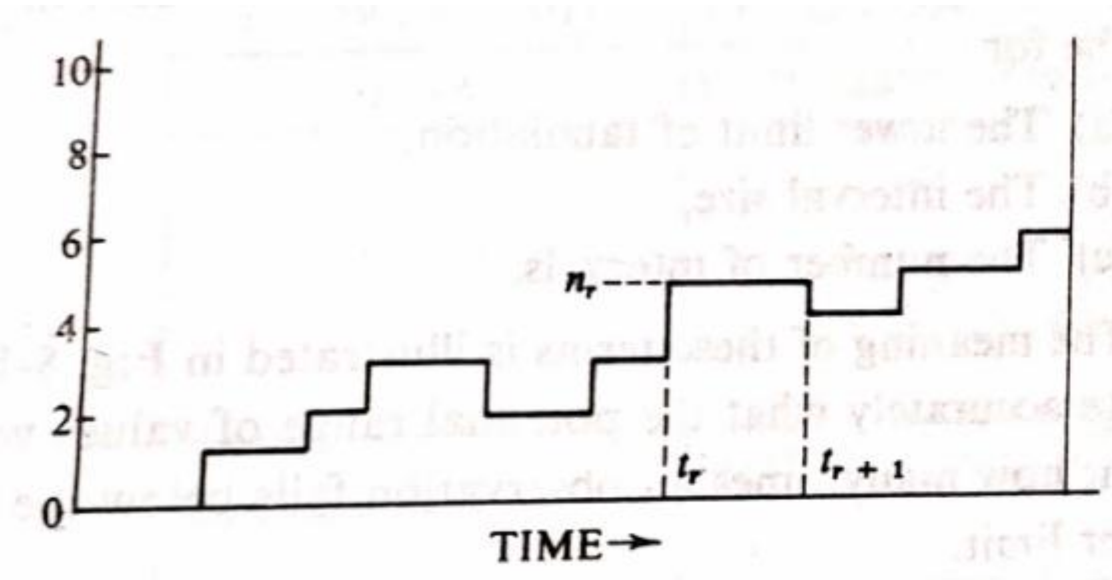


Figure: time history of busy telephone links

Above figure represents as a function of time, the number of links in a telephone system that are busy. To find average no of links in use record of currently used links and time of last change should be kept. If number of changes at time t_r to the value n_r then at the time of next change t_{r+1} ,the quantity n_r (t_{r+1}-t_r)should be calculated and accumulated to total. Therefore for run A at time T

$$A = \frac{1}{T} \sum_{r=1}^{N} n_r(t_{r+1} - t_r)$$

If there is an upper limit on the number of entities, as there was a limit on the links in telephone system, the term occupancy is used to describe the average number in use as a ratio to the maximum. Thus, if there are M links in a telephone exchange and quantity n_r is the number busy in interval t_r to t_{r+1} , the average occupancy assuming the number n_r changes N times is

$$B = \frac{1}{NM} \sum_{r=1}^{N} n_r(t_{r+1} - t_r)$$

For utilization timing information is needed and for occupancy a count of class of entities and record of count that changed last time is required.

Recording Distribution and Transit Times

To record the distribution of variable, a count is required that gives the value that the variable falls within specific intervals. A table is set which gives the location to record the values for specific interval and accumulate each count. For new observation the value is compared with the limits established for intervals and 1 is added to counter.

The tabulation intervals are uniform in size and contains

a) The lower limit
b) The interval size
c) The number of intervals

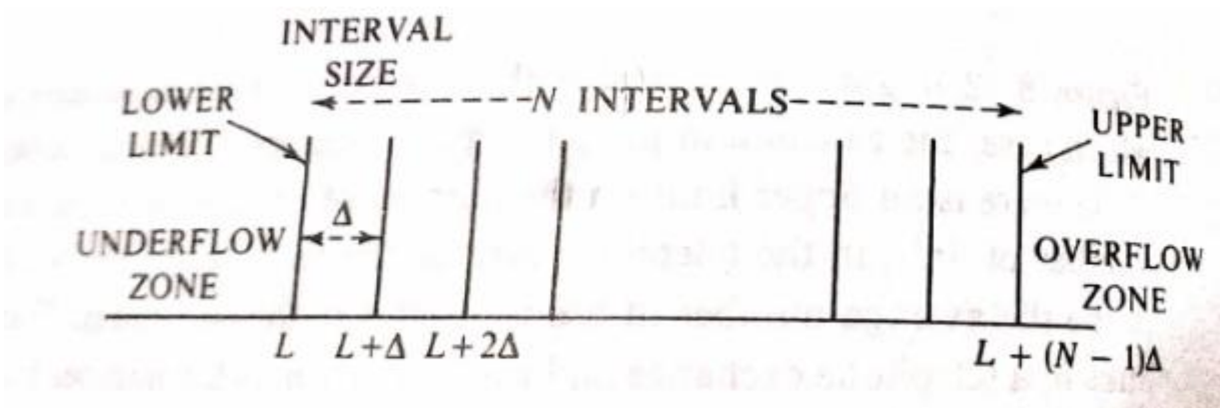


Figure: Definition of a distribution Table

It is necessary to count how many times an observation falls below the lower limit and beyond the upper limit. To calculate mean value and the standard deviation, number of observation is accumulated and the squares of the observations is added at the same time of distribution. Rach observation x_i increase a count to 1 in the counter and total x_i is Σx_i and the squared sum is $\Sigma {x_i}^2$.

To measure transit time, the clock is used in the manner of a time stamp. When and entity reaches a point from which a measurement of transit time is to start, a note of arrival time is made. Later, when the entity reaches the point at which the measurement ends a note of clock time upon arrival is made and compared with the first time to derive the elapsed interval.

General Purpose System Simulation

The system to be simulated in GPSS is described as a block diagram in which block represents the activities and lines joining the blocks indicate the sequence in which the activities can be executed. Where there is choice of activities more than one line leaves a block and the condition for the choice is stated at the block. Each block must give precise meaning. There are 48 specific blocks. Each of which represents a characteristic action of systems. The program should be written using these block diagrams.

Entities of the system depend upon the nature of the system. For eg in simulation of a communication system movement of messages, in road transportation moving of vehicles are entities. These entities are called transactions. The sequence of events in real time is reflected in movement of transactions from block to block in simulated time.

Transactions starts from generate block and stops at terminate block. Transaction is hold in block and most blocks can hold many transactions simultaneously. Transfer of transaction from one block to another occurs instantaneously at a specific time or when some change of system condition occurs.

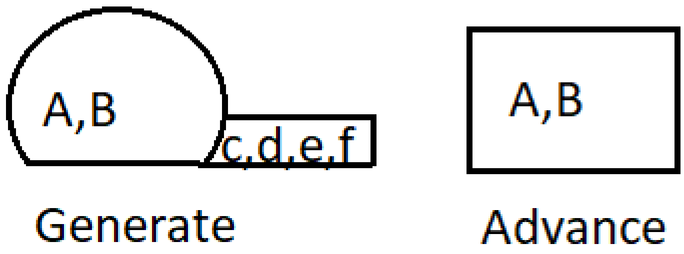
Action Times

Clock time is represented by an integral number with the interval of real time corresponding to a unit of time chosen by the program user. **ADVANCE** blocks represents the expenditure of time. The program computes an interval of time called action time for each transactions as it enters an ADVANCE block, and the transaction remain at this block for this interval of simulated time before attempting to proceed.

Another block that represents action time is **GENERATE** block that creates transaction. The action time at this block controls the interval between successive arrivals of transactions.

Action time may be fixed interval or a random variable and it can depend upon the conditions in the system in various ways. An action time is defined by giving a mean and modifier as A and B fields for the block. If modifier is zero, the action time is a constant equal to mean. If modifier is a positive number greater than or equals to mean, the action time is an integer random variable chosen from the range **mean± modifier** with equal probability of occurrence given to each number in the range.

Functions can be introduced to relate input and output variable. By specifying modifier at Advance or Generate block to be a function, the value of function controls the action time. The action time is derived by multiplying the mean by value of function.



Field A and field B in both blocks are mean and modifier. The fields c, d, e, f in generate block indicates:

c- offset (arrival time of first entity)

d- count (number of arrivals of entity)

e- priority (low and high priority of entity)

f- parameters (gives data types)

Succession of Events

The program maintains records of when each transaction in the system is due to move. It proceeds by completing all movements that are scheduled for execution at a particular instant of time. When there is more than one transactions to move, the program processes transactions in order of priority with first-come-first serve basis.

Transactions do not spend time at any block other than at an ADVANCE block. The program begins moving transaction through the block diagram until one of several circumstances arises. The transaction may enter on ADVANCE block with a non-zero action time, the program will turn its attention to other transactions in the system and return to the transaction when the action time has been expanded.

Secondly – A condition arises that the transaction is attempting to execute by entering block cannot be performed at the current time. The transaction is said to be blocked and it remains at the block it last entered. The program will automatically detect when the blocking condition is removed and will start the transaction at that time.

A third possibility is that transaction enters TERMINATE block in which case it is removed from simulation.

Fourth possibility is that a transaction may be put on a chain. When the program has moved one transaction as far it can go, it turns to other transactions that is to move at same time instant. If all such movements are complete, the program advances the clock to the time of next most imminent event and repeats the process of executing events.

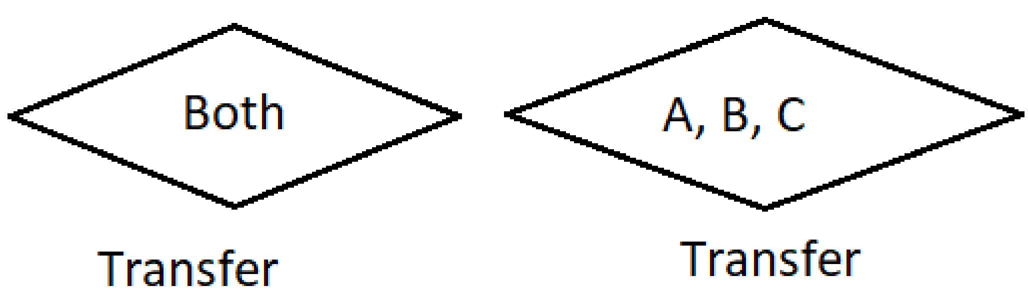
Choice of path

TRANSFER block

The TRANSFER block is used for choice of path .It allows to select the location other than the next sequential block choice is between two blocks. Selection factor in field A gives the choice to next block in field B and C. If there is no choice selection factor is left blank. An unconditional transfer is then made to next block A.

Two modes of choice

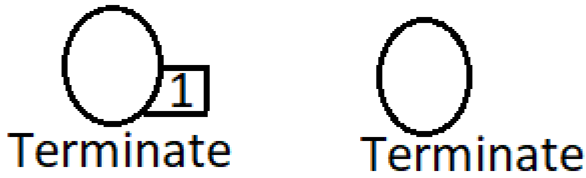
1) A random choice can be made by setting the selection factor, S. The probability of going to next block A, is then 1-S and to C is S.
2) A conditional mode, indicated by setting field A to BOTH, allows a transaction to select an alternate path depending upon existing conditions. The transaction moves to next block B if this move is possible, or to C if it is not possible. If both are impossible, transaction waits for first to become possible, giving preference to B.



Terminate Block

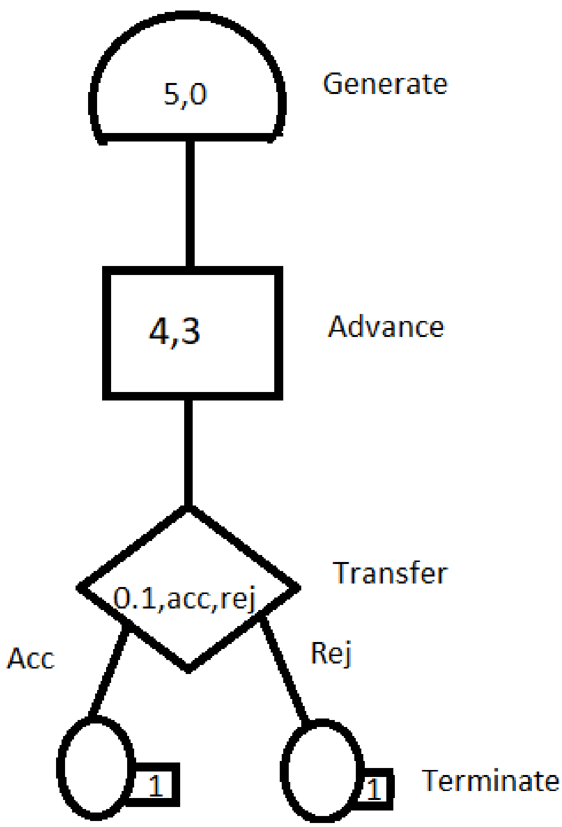
Terminate block is used to end the simulation in GPSS. There may be multiple terminate block in a program but there must be at least one terminate block with nonzero field.

Terminate block is circular shape with a tail on it with field 1.



Example

A machine tool in a manufacturing shop is turning parts at the rate of one every 5 minutes. As they are finished, the parts go to an inspector who takes 4±3 minutes to examine each one and rejects about 10% of the parts. Each part will be represented by one transaction and the time unit selected for the problem will be minute. Simulate it for 1000 parts.



Manufacturing shop model-1

Here, GENERATE block is used to represent the output of the machine by creating one transaction every five units of time. An ADVANCE block with a mean of 4 and modifier of 3 is used to represent inspection. The time spent on inspection will therefore be any one of the values 1, 2, 3, 4, 5, 6 or 7 with equal probability given to each value. Upon completion of inspection, transaction go to a TRANSFER block with a selection factor of 0.1, so that 90% of the parts go to next location called ACC to represent accepted parts and 10% go to another location called REJ to represent rejected parts. There is no history thus both locations are to TERMINATE block.

Location							Operation											Operation Field												Comments								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39

An * in column 1 results in the statement being printed in output only. A field from 2 to 6 contains location of blocks where it is necessary. The GPSS program will automatically assign sequential location as it reads the statements, so it is not necessary to assign locations. The transfer block needs to make reference to TERMINATE blocks to which it sends transactions.

The second section of coding from columns 8 to 18 contains the block type name, which must begin in column 8. From 19, a series of fields may be present each separated by commas and having no blanks. Any line following the first blank is treated as comment. The meaning of field depends upon block type.

The program accepts input in a free format in which the location field, if used, begins in column 1, but none of other fields has a fixed starting points. A single block marks the transaction from the location field to the operation field and from operation to operand field. If no location is specified an initial blank is used. The free format is very convenient hen entering input from a terminal, since it minimizes the number of characters that have to be entered. The fixed format is easier to use.

For TRANSFER block, the first field is selection factor B and C fields are exit 1 and 2 respectively. In this case exit 1 is the next sequential block, ACC can be omitted from TRANFER and TERMINATE block. Comma must be used to show that field B is missing. So TRANSFER block would be coded as TRANSFER 1, , REJ. If there is unconditional transfer, then also a comma should be included to indicate the field i.e. TRANSFER, REJ.

Program runs until a certain count is reached terminating transactions. TERMINATE block has field A that carries a number indicating by how much termination count is incremented. The number should be positive or zero. But there must be at least one non zero TERMINATE block. In our example there is one in both block thus 1 should be added in both bad and good inspection.

In last line of coding there is a control statement START. Here the START statement is set to 1000. When START is read, the program begins execution. When simulation is completed, the program prints an output report.

```
BLOCK
NUMBER   *LOC    OPERATION   A,B,C,D,E,F,G,H,I                          STATEMENT
                 SIMULATE                           COMMENTS             NUMBER
         *                                                                 1
         *  MANUFACTURING SHOP - MODEL 1                                   2
         *                                                                 3
  1              GENERATE    5                      CREATE PARTS           4
  2              ADVANCE     4,3                    INSPECT                5
  3              TRANSFER    .1,ACC,REJ             SELECT REJECTS         6
  4       ACC    TERMINATE   1                      ACCEPTED PARTS         7
  5       REJ    TERMINATE   1                      REJECTED PARTS         8
         *                                                                 9
                 START       1000                   RUN 1000 PARTS        10
                                                                          11
                             CROSS-REFERENCE
                             BLOCKS
SYMBOL          NUMBER           REFERENCES

ACC               4                  7
REJ               5                  7

        SIMULATE
*
* MANUFACTURING SHOP - MODEL 1
*
 1      GENERATE    5
 2      ADVANCE     4,3
 3      TRANSFER    .100,4,5
 4      TERMINATE   1
 5      TERMINATE   1
*
        START       1000

RELATIVE CLOCK                 5005   ABSOLUTE CLOCK          5005
BLOCK COUNTS
BLOCK  CURRENT         TOTAL
  1       0             1001
  2       1             1001
  3       0             1000
  4       0              888
  5       0              112
```

Problem input is printed first, with the locations listed from left and a sequential statement number on right. Then table of symbolic location to each symbol.

The first line of output following the listings gives the time at which simulation stop. Time is followed by listing block counts. Two numbers are shown for each block. On the left is count of how many transactions were in the block at simulation stop time and on right is a figure showing total number of transactions entering the block during simulation.

The results show that the counts at block 4 and 5 were 888 and 112 respectively showing 1000 parts inspected i.e. 88.8% accepted and 11.2% rejected.

Facilities and Storages

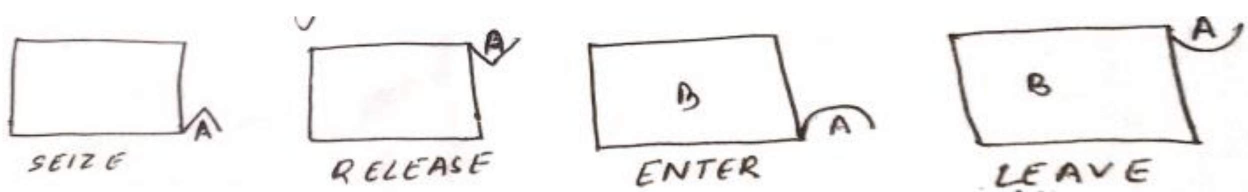
A facility is defined as an entity that can be engaged by a single transaction at a time. A storage is defined as an entity that can be occupied by many transactions at a time, up to some predetermined limit. A transaction controlling a facility can be interrupted or preempted by another transaction.

There can be many instances of each type of entity to a limit set by the program. Entities are identified by number. 0 cannot be assigned. User may assign numbers in any order or symbolic names can also be assigned.

Eg

Types of System	Transaction	Facility	Storage
Communications	Message	Switch	Trunk
Transportation	Car	Tollbooth	Road
Data processing	Record	Key Punch	Computer Memory

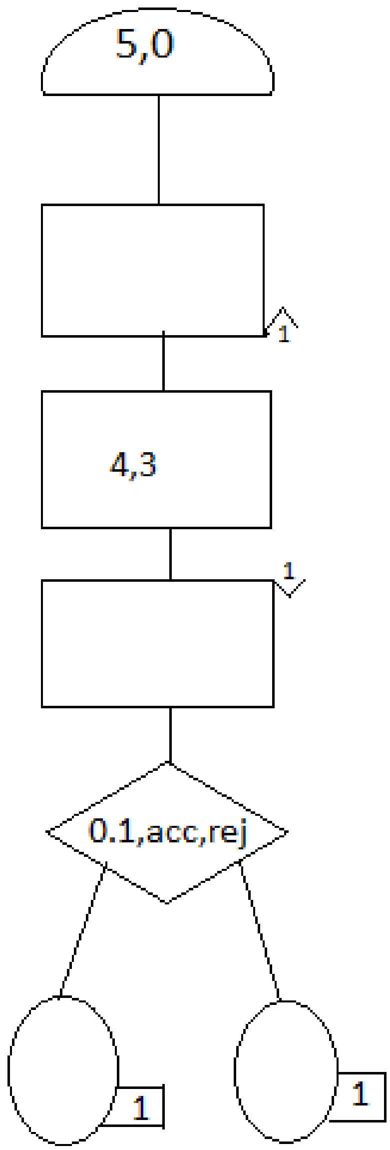
SIEZE, RELEASE, ENTER and LEAVE are used for facilities and storages.



Field A in each case indicates which facility or storage is intended, and the choice is usually marked in the flag attached to the symbols of the blocks. The SEIZE block allows a transaction to engage a facility if it is available. The RELEASE block allows the transaction to disengage facility.

Enter block allows a transaction to occupy space in a storage if available and LEAVE block allows it to give up the space. If field B of the ENTER and LEAVE block are blank, the storage content are changed by 1. If there is a number ≥1, then the contents change by that value. Number of blocks can be used between the points where facility is seized and released to simulate the actions. Similar is the case for storage.

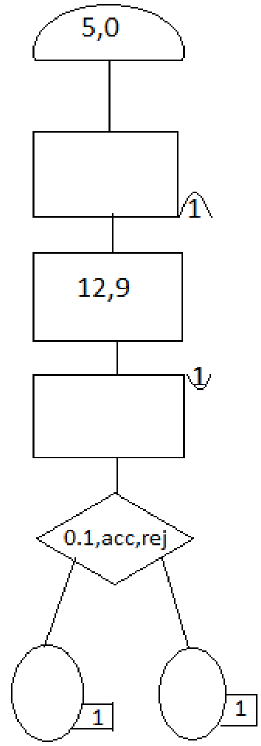
Example of manufacturing shop has average inspection time 4 and average generate rates of parts 5 minutes. There will be normally one part inspected at a time. A new part can arrive before the inspection of previous one. Thus there will be more than one transaction in ADVANCE block at one time. Assuming for only one inspector and the inspector should be represented by facility.


5,0
1
4,3
1
0.1,acc,rej
1
1

```
          * MANUFACTURING SHOP - MODEL 2
          *
 1              GENERATE    5              CREATE PARTS
 2              SEIZE       1              GET INSPECTOR
 3              ADVANCE     4,3            INSPECT
 4              RELEASE     1              FREE INSPECTOR
 5              TRANSFER    .1,ACC,REJ     SELECT REJECTS
 6        ACC   TERMINATE   1              ACCEPTED PARTS
 7        REJ   TERMINATE   1              REJECTED PARTS
          *
                START       1000           RUN 1000 PARTS

RELATIVE CLOCK          5435   ABSOLUTE CLOCK           5435
BLOCK COUNTS
BLOCK CURRENT       TOTAL
  1      1          1001
  2      0          1000
  3      0          1000
  4      0          1000
  5      0          1000
  6      0           900
  7      0           100

                ****************************************
                *                                      *
                *             FACILITIES               *
                *                                      *
                ****************************************

                                   -AVERAGE  UTILIZATION  DURING-
FACILITY  NUMBER   AVERAGE     TOTAL   AVAIL.   UNAVAIL.   CURRENT   PERCE
          ENTRIES  TIME/TRAN   TIME    TIME     TIME       STATUS    AVAILABI
    1      1000      3.995     .735                                   100.(
```

The result show inspector was busy for 73.5% of his time. If more than one inspector is available, they can be represented by storage with capacity equal to no. of inspector. SEIZE and RELEASE in the figure should be replaced by ENTER and LEAVE.

Eg Inspection time were 3 times as long as before, and there is 3 inspector, then storage should be added. STORAGE statement has location field which identifies it and operation field A has the capacity greater than or equal to 2. The difference between ENTER, LEAVE and SIEZE, RELEASE is that in facility transaction that grabs entity should free the entity but is not necessary all transactions should free storage. Entering and leaving is different actions in storage.



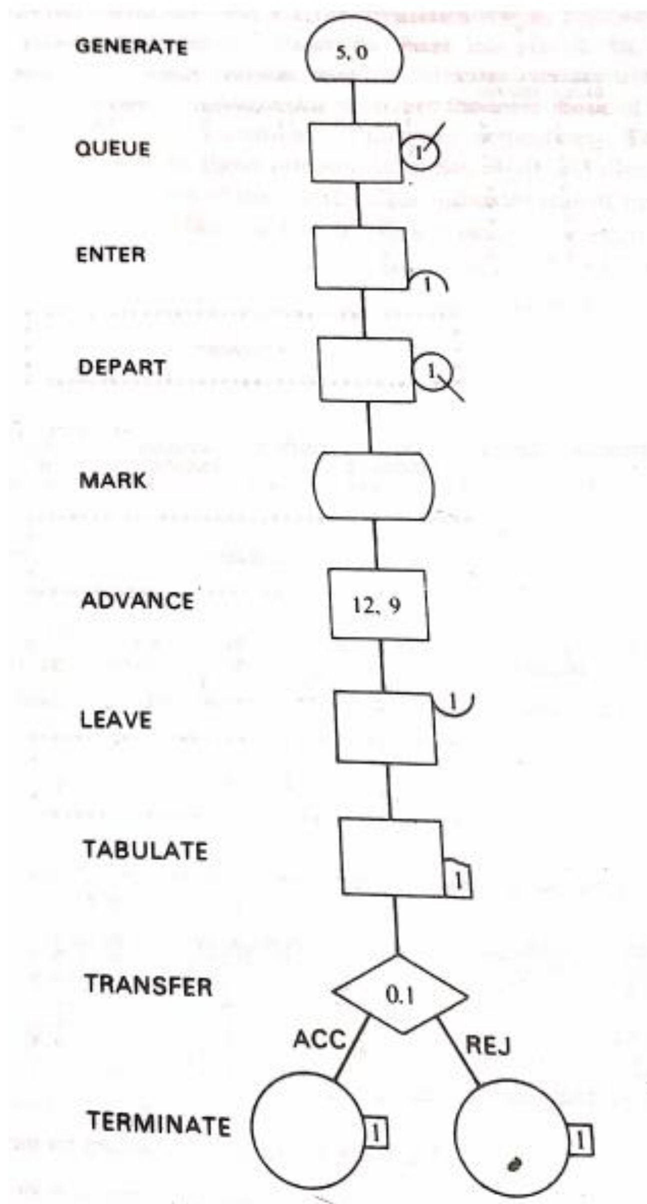

```
        * MANUFACTURING SHOP - MODEL 3
        *
1             GENERATE    5              CREATE PARTS
2             ENTER       1              GET AN INSPECTOR
3             ADVANCE     12,9           INSPECT
4             LEAVE       1              FREE INSPECTOR
5             TRANSFER    .1,ACC,REJ     SELECT REJECTS
6       ACC   TERMINATE   1              ACCEPTED PARTS
7       REJ   TERMINATE   1              REJECTED PARTS
        *
        1     STORAGE     3              NUMBER OF INSPECTORS
        *
              START       1000           RUN 1000 PARTS

RELATIVE CLOCK          5355   ABSOLUTE CLOCK          5355
BLOCK COUNTS
BLOCK CURRENT      TOTAL
  1      1          1003
  2      0          1002
  3      2          1002
  4      0          1000
  5      0          1000
  6      0           893
  7      0           107

                 ****************************************
                 *                                      *
                 *               STORAGES               *
                 *                                      *
                 ****************************************

                                                -AVERAGE  UTILIZATION  DURING-
STORAGE  CAPACITY  AVERAGE   ENTRIES  AVERAGE    TOTAL     AVAIL.     CURRENT   MAXIMUM
                   CONTENTS           TIME/UNIT  TIME      TIME       CONTENTS  CONTENTS
   1        3       2.256     1002     12.059    .752                    2         3
```

Gathering statistics

The **QUEUE, DEPART, MARK, TABULATE** block are used for gathering statistics.

When the conditions for advancing a transaction are not satisfied several transactions may be kept waiting at a block. When the conditions are favorable they are moved with first-in, first-out basis. The **QUEUE** block increases and **DEPART** block decreases the queue numbered in field A. If field B is blank the change is a unit change otherwise value of B is ≥ 1.

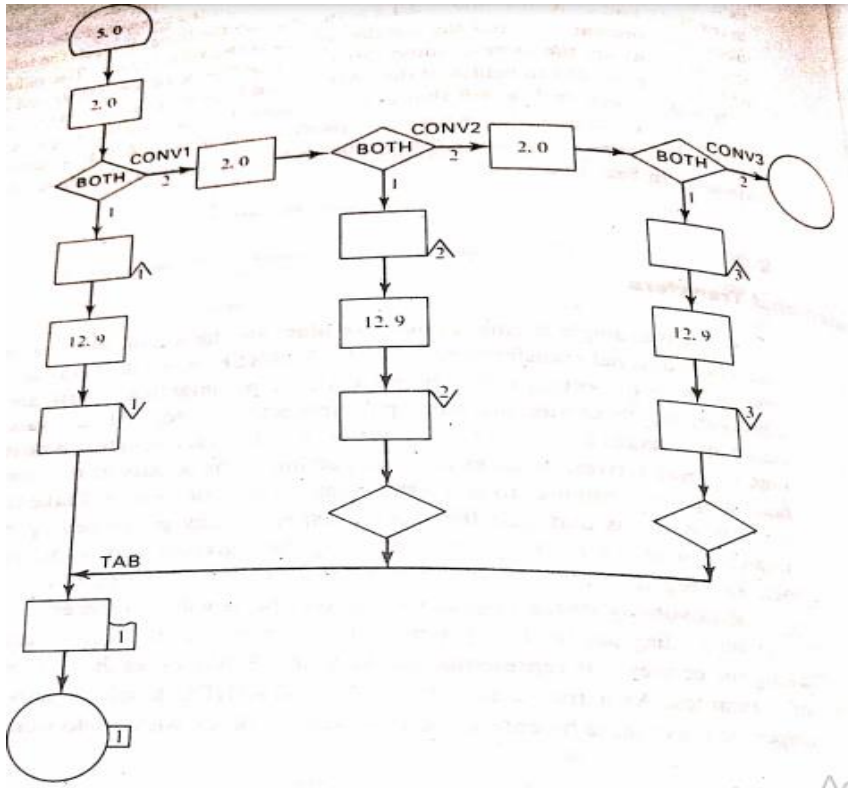
MARK and **TABULATE** gives the length of time taken by transactions to move through the system. The MARK block notes the time of arrival. The TABULATE block subtracts the time noted by MARK block from the time of arrival at TABULATE block. The transit time is entered in table whose number or name is indicated in field A of TABULATE. If transaction entering TABULATE block is not passed through MARK block, the transit time is derived by using as a base the time at which transaction was created.



```
     * MANUFACTURING SHOP - MODEL 4
     *
             GENERATE    5             CREATE PARTS
  1          QUEUE       1             QUEUE FOR AN INSPECTOR
  2          ENTER       1             GET AN INSPECTOR
  3          DEPART      1             LEAVE QUEUE
  4          MARK
  5          ADVANCE     12,9          INSPECT
  6          LEAVE       1             FREE INSPECTOR
  7          TABULATE    1             MEASURE TRANSIT TIME
  8          TRANSFER    .1,ACC,REJ    SELECT REJECTS
  9   ACC    TERMINATE   1             ACCEPTED PARTS
 10   REJ    TERMINATE   1             REJECTED PARTS
 11  *
      1      STORAGE     3             NUMBER OF INSPECTORS
      1      TABLE       M1,5,5,10     TABULATION INTERVALS
     *
             START       1000          RUN FOR 1000 PARTS

RELATIVE CLOCK         5005  ABSOLUTE CLOCK          5005
BLOCK COUNTS
BLOCK CURRENT    TOTAL    BLOCK CURRENT    TOTAL
  1      0       1000      11      0        108
  2      0       1000
  3      0       1000
  4      0       1000
  5      0       1000
  6      0       1000
  7      0       1000
  8      0       1000
  9      0       1000
 10      0       1000
         0        892
```

Conditional Transfers

Again consider the case of three inspectors and the manufactured parts are put on a conveyor which carries the parts to the inspector at intervals along the conveyor. It take 2 minutes for a part to reach to the first inspector; if it is free it takes the part for inspection else it takes further 2 minutes to reach to second inspector who will take it if its free else it passes to the part to third inspector which again takes 2 minutes otherwise they are lost. To keep the model small only the transit time of the parts are recorded and the possibility of inspector rejecting parts are ignored.


5.0
2.0
CONV1
BOTH
2
1
2.0
BOTH
CONV2
2
1
2.0
BOTH
CONV3
2
1
1
12.9
1
2
12.9
2
3
12.9
3
TAB
1
1

```
* MANUFA...
*
1              GENERATE     5              CREATE PARTS
2              ADVANCE      2              PLACE ON CONVEYOR
3              TRANSFER     BOTH,,CONV1    MOVE TO FIRST INSPECTOR
4              SEIZE        1              GET FIRST INSPECTOR
5              ADVANCE      12,9           INSPECT
6              RELEASE      1              FREE INSPECTOR
7      TAB     TABULATE     1              MEASURE TRANSIT TIME
8              TERMINATE    1
*
9      CONV1   ADVANCE      2              PLACE ON CONVEYOR
10             TRANSFER     BOTH,,CONV2    MOVE TO SECOND INSPECTOR
11             SEIZE        2              GET SECOND INSPECTOR
12             ADVANCE      12,9           INSPECT
13             RELEASE      2              FREE INSPECTOR
14             TRANSFER     ,TAB
*
15     CONV2   ADVANCE      2              PLACE ON CONVEYOR
16             TRANSFER     BOTH,,CONV2    MOVE TO THIRD INSPECTOR
17             SEIZE        3              GET THIRD INSPECTOR
18             ADVANCE      12,9           INSPECT
19             RELEASE      3              FREE INSPECTOR
20             TRANSFER     ,TAB
*
21     CONV3   TERMINATE
*
       1       TABLE        M1,5,5,10      TABULATION INTERVALS
*
               START        10,NP          INITIALIZE WITH TEN PARTS
               RESET
               START        1000           MAIN RUN

RELATIVE CLOCK           5588   ABSOLUTE CLOCK            5660
```

BLOCK COUNTS

BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL
1	0	1117	11	0	343	21	0	118
2	0	1117	12	0	343			
3	0	1118	13	0	344			
4	0	406	14	0	344			
5	1	406	15	1	369			
6	0	406	16	0	368			
7	0	1000	17	0	250			
8	0	1000	18	0	250			
9	0	712	19	0	250			
10	0	712	20	0	250			

FACILITIES

FACILITY	NUMBER ENTRIES	AVERAGE TIME/TRAN	-AVERAGE UTILIZATION DURING- TOTAL TIME	AVAIL. TIME	UNAVAIL. TIME	CURRENT STATUS	PERCENT AVAILABILITY
1	407	11.885	.865				100.0
2	344	11.956	.736				100.0
3	250	12.244	.547				100.0

Program Control Statement

The first statement of GPSS input is a control statement SIMULATE in operation field. Without this statement problem will be assembled but not in use.

Simulation run can be restart or stop and can also repeat changing value to some extent. When GPSS simulation run is finished, the program does not destroy the model immediately. It looks input following the START statements. Input following the START statement can change the model. A STORAGE statement can be inserted giving new value. The model can also be changed by changing existing block or adding new blocks. When the desired changes is made the model reruns with another START statement.

RESET is another control statement. It is used to wipe out all the statistics gathered. It will leave the system loaded with transactions to gather statistics in second run. The output of first run is then not of interest and is used as NP in field B of START. One start statement has run the model for 10 completed transactions. A RESET statement has wipe out the statistics in that run. The second START statement has restarted the simulation from the point of 10^{th} transaction and has continued for 100 transactions.

The RESET statement also sets relative clock to zero. Absolute clock gives the time since the run began. Relative clock gives the time since the last reset statement. If reset is not used both time are same.

CLEAR statement wipes out the statistics and transactions in the system, so that rerun starts the simulation from the beginning. The CLEAR statements return the model to its initial state. It does not reset the random number generator seeds. The sequence of statements

START

CLEAR

START

would run the same problem twice but second run would use a different sets of random numbers.

JOB wipes out the entire model preceding the statement and proceed with following problem.

END terminates all simulation. SIMULATE statement appears only once even if there are multiple jobs.

SIMSCRIPT

SIMSCRIPT language is used for simulating discrete system. SIMSCRIPT uses entities and attributes. It distinguishes between temporary and permanent entities and attributes. Temporary entities are created and destroyed during execution of a simulation and permanent entities remain during the run. Temporary entities are placed on sets. User can define sets for entering and removing entities into and from sets.

Activities are considered as events that occurs instantaneously. Each event is described by an event routine. Each event routine is given name and each programmed as a subroutine. Distinction is made between endogenous and exogenous event. An endogenous event occur by a scheduling statement in some event routine. Exogenous event requires the reading of data supplied by user. Among the data is the time at which the event is to occur. Event routines are needed to execute the changes that result when an external events becomes due for execution. Automatic initialization procedure of SIMSCRIPPT is to prepare the first exogenous event from each data set. The concept used in SIMSCRIPT are:

Entities

Permanent

Temporary

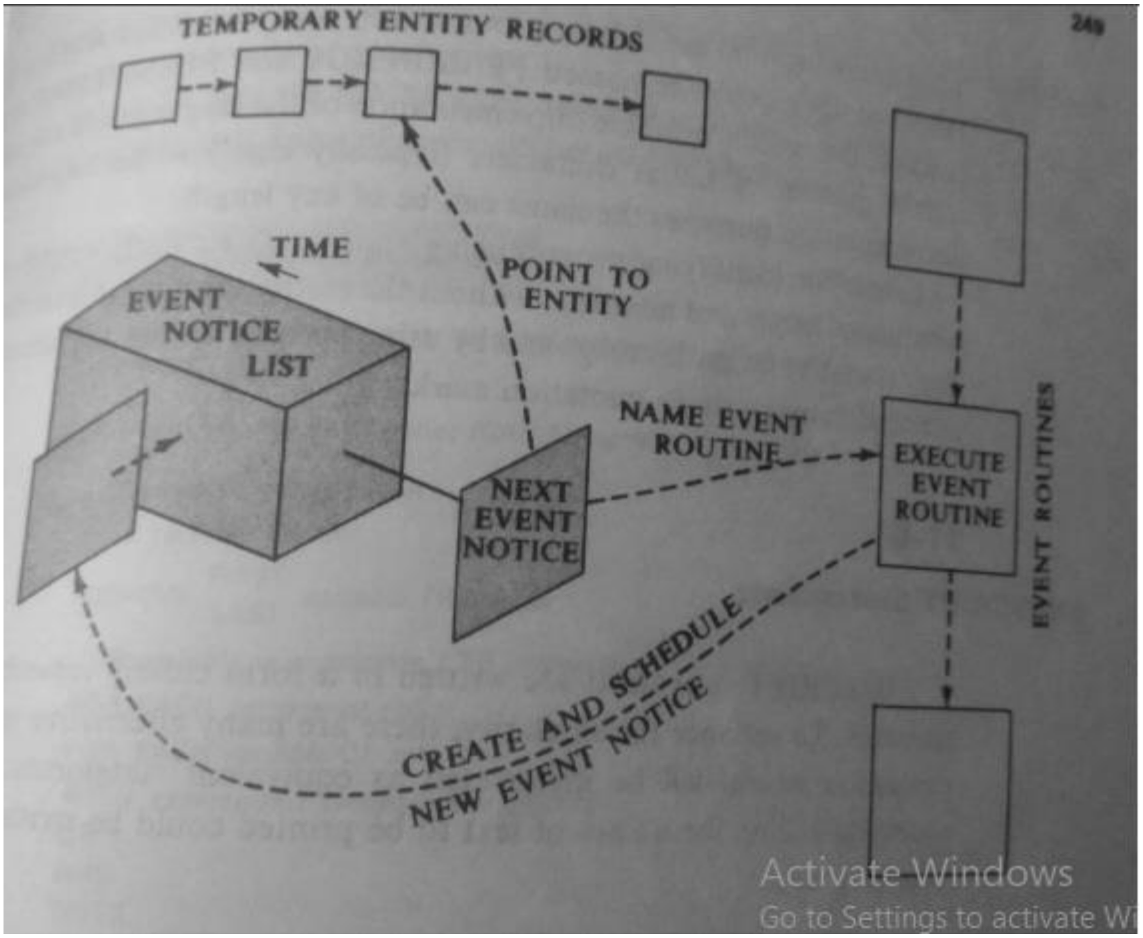
Sets

Event Routines

Organization of a SIMSCRIPT

Event routines are closed routines. Thus some means must be provided for transferring control between them. Event routines are used to transfer controls. Event notices are created when it is determined that an event is scheduled. An event notice exists for endogenous event scheduled to occur. They record time of arrival of event and the event routine that executes the event.

When all events that can be executed at a particular time have been processed, the clock is updated to thee time of next event notice and control is passed to event routine identified by notice. This process is automatic and need not to be scheduled.



If the event executed by a routine results in another event, either at the current clock time or in future, the routine must create a new event notice and file it with other notices. E.g. in telephone call the connection call and disconnect call are linked. Thus event notice should be scheduled for disconnection by the routine that connects call.

A series of exogenous event statements are created, one for each event. These statement are similar to event notices and they give the arrival time of event and identify the exogenous routine to execute event.

Name and Labels

All the entities and its attributes must be given name. Names may consist of any combination of letters and digits provided. Example AGE and EDUCATION of a PERSON can be named as PERSON.AGE and PERSON.EDUCATION.

Labels for identifying programming statements similarly consist of any combination of letters and numbers without restriction that at least one be a letter.

SIMSCRIPT Statements

SIMSCRIPT is closely related to English language. Example a statement calling n lines of text to be printed could be written as:

PRINT n LINES AS FOLLOWS

PRINT n LINES THUS

If the values of some variables are to be included in the text, the statement can be written as

PRINT n LINES WITH X and Y LIKE THIS

The symbols X and Y represent the variables or expressions to be printed. Numbers are printed as *** and for decimals **.*.

Comments can be included with in two single quote and can be concluded by another two single closing quote. For long statements closing quote can be avoided.

E.g. SIMSCRIPT Statements

CREATE temporary entity or event notice (CALLED variable)

DESTROY

SCHEDULE AN event notice (CALLED variable) IN/AT expression

Defining the Telephone System Model

Telephone lines are permanent entities and each needs an attribute indicating the line is busy or not. TLINE is the name given to telephone lines and STATE is its attributes. It is integer variable that takes 0 and 1, to represent free or busy state. Calls are temporary entities with two attributes: origin and destination.

Two system variable is needed to carry the maximum number of links and the number currently in use. The maximum number is initialized from data read into program. The mean arrival time, the mean call length, simulation time and variables needed are defined for these quantities. Other variables will be used to collect statistics.

The preamble section includes

```
1   ''   TELEPHONE SYSTEM - MODEL 1
2   PREAMBLE
3   NORMALLY,MODE IS INTEGER
4   EVENT NOTICES INCLUDE ARRIVAL AND CLOSING
5        EVERY DISCONNECT HAS A DIS.CALL
6   TEMPORARY ENTITIES
7        EVERY CALL HAS AN ORIGIN AND A DESTINATION
8   DEFINE LINKS.IN.USE, MAX.LINKS, BLOCKED, BUSY, FINISHED
9        AND STOP.TIME AS VARIABLES
10  DEFINE INTER.ARRIVAL.TIME AND MEAN.LENGTH AS REAL VARIABLES
11  PERMANENT ENTITIES
12       EVERY TLINE HAS A STATE
13       DEFINE SECS TO MEAN /60 MINUTES
14  END
```

Figure 11-2. Telephone system - model 1, preamble.

Line 1 is a comment line. It is not essential to use a coat mark at the end of comment. Enter arrival time and mean call length is defined as real variables.

Lines 4 and 5 give event notices. Name of event notices must be same as event routines. The routine are

ARRIVAL, CLOSING and DISCONNECT. ARRIVAL and CLOSING donot need attributes thus they are included in event header. The DISCONNECT routine needs which particular call is disconnected thus the event notice need an attribute to identify call.

Lines 6 and 7 are temporary entities representing calls. It has two attributes origin and destination. Lines 8, 9 and 10 defines the variables as integers and real. Lines 11 and 12 define the permanents entities representing the telephone lines. Line 13 defines time unit of second and 14 is END statement.

Referencing variable

Single variables are references by using their names. Permanent entities are by represented by arrays with an array for each attribute, identified by the attribute name. Individual entities are identified by an integer number which is an index to array. An array defined by user as a system variable is similarly entered by using an integer index. E.g. Reference to state of telephone line is STATE (I), where I is integer or an expression that can be evaluated as an integer. If expression is used it can also be indexed. E.g. STATE (ORIGIN(CALL)))

It refers to state that identifies origin attribute of event notice call.

For temporary entities. E.g.

CREATE A CALL or DESTROY A CALL

Since many calls can be created i.e. blocks can be created a pointer is used to give the location of block. For every type of temporary entity or event notice, the system automatically reserves a location for holding a pointer that has the same name as the entity or notice. E.g. the command CREATE CALL puts the pointer to created record in a location called CALL.

The Main Routine

```
MAIN
READ N.TLINE, MAX.LINKS, INTER.ARRIVAL.TIME, MEAN.LENGTH AND STOP.TIME
CREATE EVERY TLINE
IF N.TLINE < 2*MAX.LINKS + 2
      PRINT 1 LINE THUS
TOO FEW LINES SPECIFIED. SIMULATION ABANDONED.
      STOP
ELSE
SCHEDULE AN ARRIVAL NOW
SCHEDULE A CLOSING IN STOP.TIME MINUTES
START SIMULATION

''    REPORT TO BE WRITTEN AT END OF SIMULATION

      PRINT 1 LINE THUS
IMULATION OF TELEPHONE SYSTEM - MODEL 1
      SKIP 3 OUTPUT LINES
      PRINT 5 LINES WITH N.TLINE, MAX.LINKS, INTER.ARRIVAL.TIME,
            MEAN.LENGTH, STOP.TIME THUS
 NUMBER OF LINES                       *
 NUMBER OF LINKS                       *
 MEAN INTER-ARRIVAL TIME             *.* /SECONDS
 MEAN CALL LENGTH                    *.* /SECONDS
 SIMULATED TIME                        * /MINUTES
      SKIP 2 OUTPUT LINES
      PRINT 3 LINES WITH FINISHED, BUSY, BLOCKED THUS
 CALLS PROCESSED                       *
 BUSY CALLS                            *
 BLOCKED CALLS                         *
      STOP
END
```

Output

```
SIMULATION OF TELEPHONE SYSTEM - MODEL 1


    NUMBER OF LINES                  50
    NUMBER OF LINKS                  10
    MEAN INTER-ARRIVAL TIME        12.0 /SECONDS
    MEAN CALL LENGTH              120.0 /SECONDS
    SIMULATED TIME                  200 /MINUTES


    CALLS PROCESSED                 988
    BUSY CALLS                      218
    BLOCKED CALLS                    76
```

The arrival routine

```
EVENT ARRIVAL
    CREATE CALL
    LET ORIGIN(CALL) = RANDI.F(1,N,TLINE,1)
    UNTIL STATE(ORIGIN(CALL)) = 0,
        LET ORIGIN(CALL) = RANDI.F(1,N,TLINE,1)
    LET DESTINATION(CALL) = RANDI.F(1,N,TLINE,1)
    UNTIL DESTINATION(CALL) NE ORIGIN(CALL),
        LET DESTINATION(CALL) = RANDI.F(1,N,TLINE,1)
    IF STATE(DESTINATION(CALL)) = 0 AND LINKS.IN.USE < MAX.LINKS,
        LET STATE(ORIGIN(CALL)) = 1
        LET STATE(DESTINATION(CALL)) = 1
        ADD 1 TO LINKS.IN.USE
        SCHEDULE A DISCONNECT GIVEN CALL
             IN EXPONENTIAL.F(MEAN.LENGTH,1) SECS
    ELSE
        IF LINKS.IN.USE = MAX.LINKS,
             ADD 1 TO BLOCKED
        ELSE
             ADD 1 TO BUSY
        ALWAYS
        ADD 1 TO FINISHED
        DESTROY CALL
    ALWAYS
    SCHEDULE AN ARRIVAL IN EXPONENTIAL.F(INTER.ARRIVAL.TIME,1) SECS
    RETURN
END
```

Timing Routine

- The initial arrival of a call will be followed by two event notices. One for next arrival and one for disconnect of first arrival. They could occur in either order.
- Execution of ARRIVAL routine will produce two more event notices.
- Whenever an event routine takes control, the program automatically returns to a section of program that selects the next event notices.
- The clock is updated to the time of that event and control is passed to event routine associated with event notices.

The Disconnect Event

```
EVENT DISCONNECT GIVEN CALL
    LET STATE(ORIGIN(CALL)) = 0
    LET STATE(DESTINATION(CALL)) = 0
    DESTROY CALL
    SUBTRACT 1 FROM LINKS.IN.USE
    ADD 1 TO FINISHED
    RETURN
END
```

The closing event

```
EVENT CLOSING
    CANCEL THE ARRIVAL
    DESTROY THE ARRIVAL
    RETURN
END
```

Analysis of Simulation Output

1. **Estimation method:**
This method is used to estimate parameters from observation on random variables. A random variable is drawn from an infinite population that has stationary probability distribution with a finite mean μ and variance σ^2. This means the population distribution is not affected by the number of sample already made nor does it change with time. Random variables are mutually independent. These variables are called i.i.d variables i.e. independent and identically distributed.

Central limit theorem can be used for i.i.d data. This theorem states that the sum of n iid variables drawn from a population that has mean of μ and variance σ^2 is approximately distributed as a variable with mean n μ and n σ^2.

Any normal distribution can be transformed into a standard normal distribution that has a mean of 0 and variance of 1. Let x_i (i=1, 2,…………n) be n i.i.d random variables. Now by central limit theorem

$$z = \frac{\sum_{i=1}^{n} x_i - n\mu}{\sqrt{n}\,\sigma}$$

- **In terms of sample mean $\bar{x}$**

$$z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$$

Where,

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

It can be shown to be a consistent estimator for the mean of population from the sample drawn. Sample mean is the sum of random variable, thus it is also a random variable. Thus confidence interval is to be established.

The probability density function of the standard normal variate is given in figure:

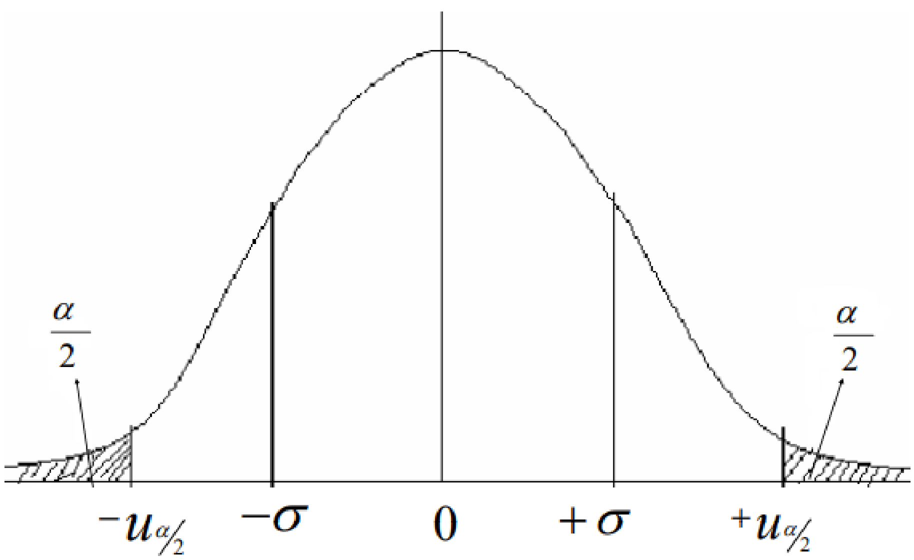


Fig:Probability density function of standard normal variate

The integral from -∞ to a value u is the probability that z is less than or equal to u. the integral is denoted by φ(u) and its table is widely available.

Let the value of u is chosen so that φ(u) =1-α/2 where α is some constant less than 1 and this value u is denoted by $u_{\alpha/2}$. the probability that z is greater than $u_{\alpha/2}$ is then α/2. The normal distribution is symmetric about its mean. So, the probability that z is less - $u_{\alpha/2}$ is also $\alpha/2$. The probability that z lies between - $u_{\alpha/2}$ and $u_{\alpha/2}$ is 1-α. i.e.

$$\text{Prob}\{-u_{\alpha/2} \le z \le u_{\alpha/2}\} = 1-\alpha$$

In terms of sample mean, this probability statement can be written as:

$$\text{Prob}\{\bar{x} + \frac{\sigma}{\sqrt{n}} u_{\alpha/2} \ge \mu \ge \bar{x} - \frac{\sigma}{\sqrt{n}} u_{\alpha/2}\} = 1-\alpha$$

The constant 1- α is the confidence level and the confidence interval is

$$\bar{x} \pm \frac{\sigma}{\sqrt{n}} u_{\alpha/2}$$

The size of confidence interval depends upon the confidence level chosen. For confidence level 90% $u_{\alpha/2}$ is 1.65. The statement then says that u will be covered by the confidence interval x ±1.65 σ √n with probability 0.9 i.e. if the experiment is repeated many times the confidence interval can be expected to cover the value u on 90% of repetitions.

2. Simulation Run Statistics

Simulation run statistics is performed for the problem that arise from simulation runs. We will explain it with specific examples. Consider a single-server system in which the arrivals occur with a Poisson distribution and the service time has an exponential distribution. Suppose the study objective is to measure the mean waiting time, defined as the time entities spend waiting to receive service and excluding the service time itself. This system is commonly denoted by M/M/1 which indicates; first , that the inter-arrival time is distributed exponentially; second that the service time is distributed exponentially; and, third, that there is one server. The M stands for Markovian, which implies an exponential distribution. In a simulation run, the simplest approach is to estimate the mean waiting time by accumulating the waiting time of n successive entities and dividing by n. This measure, the sample mean, is denoted by $x(n)$, its value depends upon the no of observations taken. If x_i (i=1,2,….,n) are the individual waiting times(including the value 0 for those entities that do not have to wait), then

$$\bar{x}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Here waiting time are not independent they are auto correlated. In a simulation run the sample mean of auto correlated data can be shown to approximate a normal distribution as sample size increases. Above equation can used to estimate the mean of auto correlated data is not related to the population variance by simple expression

Whenever a waiting line forms, the waiting time of each entity on the line clearly depends upon the waiting time of its predecessors. Any series of data that has this property of having one value affect other values is said to be auto correlated. The sample mean of auto correlated data can be shown to approximate a normal distribution as the sample size increases. The equation

$$\bar{x}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

remains a satisfactory estimate for the mean of auto correlated data. A simulation run is started with the system in some initial state, frequently the idle state, in which no service is being given and no entities are waiting. The early arrivals then have a more than normal probability of obtaining service quickly, so a sample mean that includes the early arrivals will be biased. For a given sample size starting from a given initial condition, the sample mean distribution is stationary; but , if the distributions could be compared for different sample sizes, the distribution would be slightly different. The following figure is based on theoretical results, which shows how the expected value of sample mean depends upon the sample length, for the M/M/1 system, starting from an initial empty state, with a server utilization of 0.9.

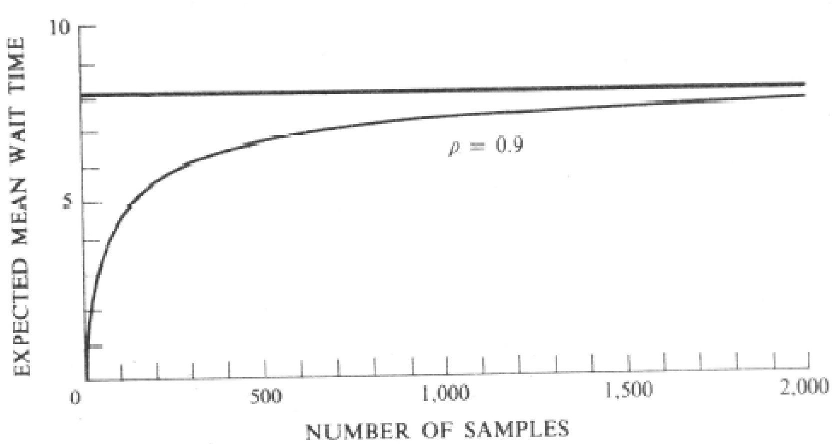


Figure 14-2. Mean wait time in M/M/1 system for different sample sizes.

Replication of Runs

The precision of results of a dynamic stochastic can be increased by repeating the experiment with different random numbers strings. For each replication of a small sample size, the sample mean is determined. The sample means of the independent runs can be further used to estimate the variance of distribution. Let X ij be the ith observation in jth run, then the sample mean and variance for the jth run are:

$$\overline{x_j}(n) = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}[x_{ij} - \overline{x_j}(n)]^2$$

When we have similar means and variances for m independent measurements, then by combining them, the mean and variance for the population can be obtained as:

$$\bar{x} = \frac{1}{p}\sum_{j=1}^{p}\overline{x_j}(n)$$

$$s^2 = \frac{1}{p}\sum_{j=1}^{p} s_j^2(n)$$

The following figure shows the result of applying the procedure to experimental results for the M/M/1 system.

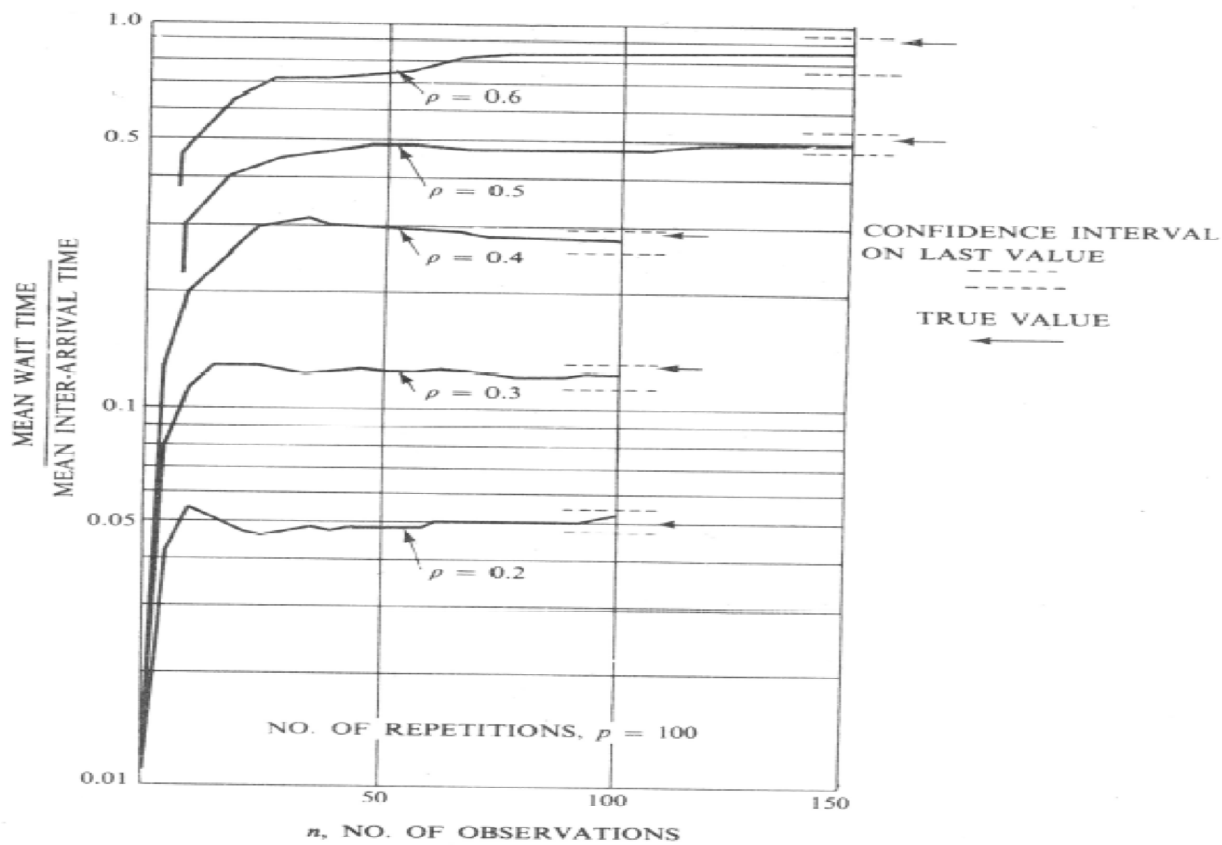


Figure 14-3. Experimentally measured wait time in M/M/1 system for different sample sizes.

This variance can further be used to establish the confidence interval for p-1 degrees degree of freedom. The length of run of replications is so selected that all combined it comes to the sample size N. i.e. p.n=N. By increasing the number of replications and shortening their length of run, the confidence interval can be narrowed. But due to shortening of length of replication the effect of starting conditions will increase. The results obtained will not be accurate, especially when the initialization of the runs is not proper. There is no established procedure of dividing the sample size N into replications.

However, it is suggested that the number of replications should not be very large, and that the sample means should approximate a normal distribution.

Elimination of Initial Bias

- Two general approaches can be taken to remove the bias: the system can be started in a more representative state that the empty state, or the first part of the simulation can be ignored.

- The ideal situation is to know the steady state distribution for the system, and select the initial condition from that distribution.
- In the study previously discussed, repeated the experiments on the M/M/1 system, supplying an initial waiting line for each run, selected at random from the known steady state distribution of waiting line.
- The more common approach to removing the initial bias is to eliminate an initial section of the run.
- The run is started from an idle state and stopped after a certain period of time.
- The run is then restarted with statistics being gathered from the point of restart.
- It is usual to program the simulation so that statistics are gathered from the beginning, and simply wipe out the statistics gathered up to the point of restart.
- No simple rules can be given to decide how long an interval should be eliminated.
- The disadvantage of eliminating the first part of a simulation run is that the estimate of the variance, needed to establish a confidence limit, must be based on less information.
- The reduction in bias, therefore, is obtained at the price of increasing the confidence interval size.