Chapter 2

Simulation of Continuous and Discrete Systems

2.1. Continuous System Models

When continuous system is modeled mathematically, the variables of model representing the attribute of system are controlled by continuous functions. The distributed lag model is an example of a continuous model. Since in continuous system, the relationship between variables describe the rate at which the value of variable change, these system consist of differential equations.

Continuous system simulation uses the notation of differential equation to represent the change on the basic parameter of the system with respect to time. Hence the Mathematical model for continuous system simulation is usually represented by differential and partial differential equations.

2.1.1. Differential Equations

An example of a linear differential equation with constant coefficients to describe the wheel suspension system of an automobile can be given as:

$$\boldsymbol{M\ddot{x}} + \mathrm{K}\boldsymbol{x} + \mathrm{D}\boldsymbol{\dot{x}} = \boldsymbol{KF(t)}$$

Here the dependent variable x appears together with first and second derivatives single dot and double dot respectively.

The simple differential equation can model the simplest continuous system and they can have one or more linear differential equation with constant coefficients. It is then often possible to solve the model without using simulation technique i.e. we can solve such equations using analytical methods as (we have done in Numerical methods)

However when non linearity involves into the model, it may be impossible or at least very difficult to solve such model without simulation.

2.1.2. Partial Differential Equations

When more than one independent variable occurs in a differential equation the equation is said to be partial differential equations. It can involve the derivatives of the same dependent variable with respect to each of the independent variable.

Differential equations both linear and nonlinear occur frequently in scientific and engineering studies. The reason for this is that most physical and chemical process involves rates of change, which require differential equation to represent their mathematical descriptions. Since partial deferential equation can also represent a growth rate, continuous model can also be applied to the problems of a social or economic nature.

A first-order partial differential equation with $n$independent variables has the general form

$$F\left(x_1, x_2, \ldots, x_n, w, \frac{\partial w}{\partial x_1}, \frac{\partial w}{\partial x_2}, \ldots, \frac{\partial w}{\partial x_n}\right) = 0,$$

where $w = w(x_1, x_2, \ldots, x_n)$ is the unknown function and $F(\ldots)$ is a given function.

Example 1: Laplace Equation

$$\frac{\partial^2 w}{\partial t^2} - \frac{\partial^2 w}{\partial x^2} = 0,$$

2.2. Analog Computers

Before the general availability of digital computers, analog computers were used for the simulation of the discrete system simulation as well continuous system simulation.

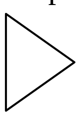
There existed devices whose behavior is equivalent to a mathematical operation such as addition, integration, etc and putting together combinations of such devices in a manner specified by a mathematical model of a system allowed the system to be simulated.

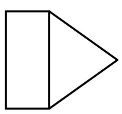
These computers were primarily used for solving differential equation and most used version of the computes was the direct current amplifiers.

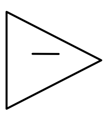
Voltage in computers is equated to mathematical variables.

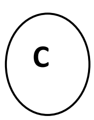
Accuracy of these computers is low fluctuations in the voltage level cannot be controlled beyond a limit:

Some of the components used in the analog computers are:

Summers 

Integrator 

Inverter 

Scalar 

Example: Simulation of wheel suspension system

Mx”+Dx’+Kx=KF(t)

Mx”=KF(t)-Dx’-kx

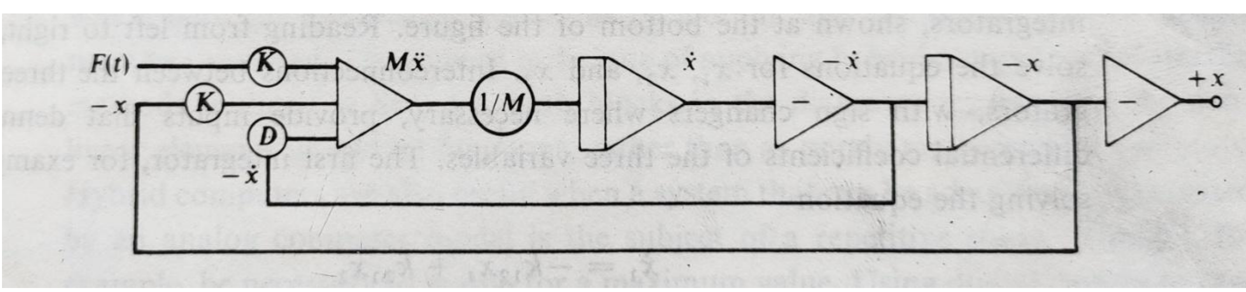


Figure: Analog computer representing Automobile Suspension System

2.3. Hybrid Computers

The scope of analog computers has been considerably extended by developments in solid-logic electronic devices. Analog computers always used a few non-linear elements such as multipliers or function generators. Originally, such devices were expensive to make. Solid logic devices, in addition to improving the design and performance of operational amplifiers, have made such nonlinear devices cheaper and easier to obtain. They have also extended the range of devices. Among the elements that can easily be associated with analog computers are circuits that carry out the logical operations of Boolean algebra, store values for later use, compare values, and operate switches for controlling runs.

The term ***hybrid computer*** has come to describe combinations of traditional analog-computer elements, giving smooth, continuous outputs, and elements carrying out such nonlinear, digital operations as storing values, switching, and performing logical operations. Originally, the term had the connotation of extending analog-computer capabilities, usually by the addition of special-purpose, and often specially constructed devices. More recently, few purely analog computers are built. Instead, computers with large numbers of standard, nonlinear elements are readily available.

Hybrid computers may be used to simulate systems that are mainly continuous, but do, in fact, have some digital elements-for example, an artificial satellite for which both the continuous equations of motion, and the digital control signals must be simulated. Hybrid computers are also useful when a system that can be adequately represented by an analog computer model is the subject of a repetitive study.

2.4. Digital analog simulators

To avoid the disadvantages of analog computers, many digital computer programming language have been written to produce digital-analog simulators. They allow or facilitate a continuous model to be programmed on a digital computer in essentially the same way as it is solved on analog computer. The language contains micro instructions that carry the action of addition, integration and sign changer. A program is written to link together these micro instructions in the same way as operational amplifiers are connected in analog computer. Since more powerful digital computer and programming language have been developed for this purpose of simulating continuous system on digital computer, the digital-analog simulators are now in extensive use.

2.5 Feedback Systems

A significant factor in the performance of many systems is that a coupling occurs between the input and output of the system. The term feedback is used to describe the phenomenon. A home heating system controlled by a thermostat is a simple example of a feedback system. The system has a furnace whose purpose is to heat a room, and the output of the system can be measured as room temperature. Depending upon whether the temperature is below or above the thermostat setting, the furnace will be turned on or off, so that information is being fed back from the output to the input. In this case, there are only two state, either the furnace is on or off.

An example of a feedback system in which there is continuous control is the aircraft system. Here, the input is a desired aircraft heading and the output is the actual heading. The gyroscope of the autopilot is able to detect the difference between the two headings. A feedback is established by using the difference to operate the control surface, since change of heading will then affect the signal being used to the heading. The difference between the desired heading θ and actual heading θ_0 is called the error signal, since it is a measure of the extent to which the system deviates from the desired condition. It is denoted byε.

The feedback in the autopilot is said to be negative feedback. The more the system output deviates from the desired from value stronger is the force to drive it back.

Feedback System Example: Autopilot System

Autopilot is an automatic aircraft control system. A gyroscope in the autopilot detects the difference between the actual heading (Heading is the relative horizontal orientation of the aircraft relative to magnetic North) and the desired heading. It sends a signal to move the control surfaces. In response to the control surface movement, the airframe steers towards the desired heading.

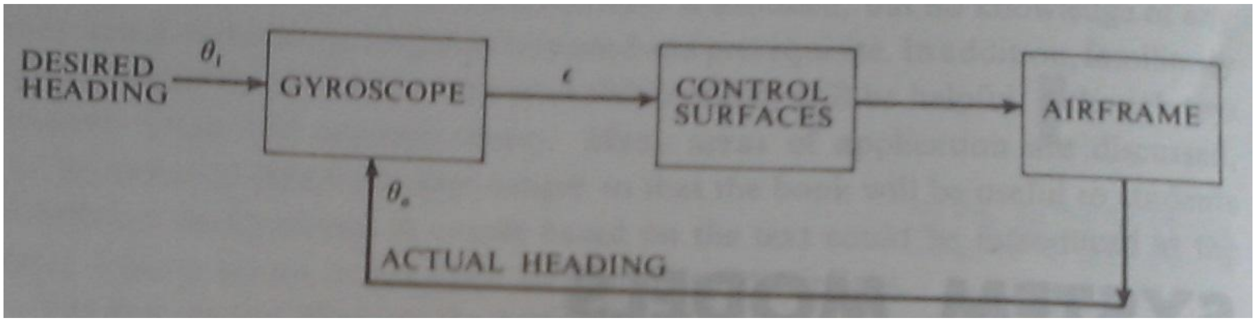


Fig: Autopilot system

In order to simulate the action of autopilot, we first construct a mathematical model of the aircraft system. The error signal, ε, has been defined as the difference between the desired heading, or output, θ_i, and the actual heading, θ_0. We therefore have the following identity:

$$\varepsilon = \theta_i - \theta_0 \qquad (1)$$

We assume the rudder (a flat, movable piece usually of wood or metal that is attached to a ship, boat, airplane, etc., and is used in steering) is turned to an angle proportional to the error signal, so that the force changing the aircraft heading is proportional to the error signal. Instead of moving the aircraft sideways, the force applies a torque which will turn the aircraft.

$$\text{Torque} = K\varepsilon - D\dot{\theta}_0 \qquad (2)$$

Where K and D, are constants, the first term on the right hand side is the torque produced by the rudder, and the second is the viscous drag.

The torque is also given by the product of inertia on body of aircraft and second derivative of the heading,

$$\text{Torque} = I\ddot{\theta}_0 \qquad (3)$$

From equation (1), (2), and (3) we get,

$$I\ddot{\theta}_0 + D\dot{\theta}_0 + K\theta_0 = K\theta_i \ (4)$$

Dividing both sides by I, and making the following substitutions in equation (4)

$2\xi\omega$ = D/I, $\omega^2 = k/I$

$$\ddot{\theta}_0 + 2\xi\omega\dot{\theta}_0 + \omega^2\theta_0 = \omega^2\theta_i \qquad (5)\ \ (\xi \text{ is damping factor})$$

This is a second order differential equation.

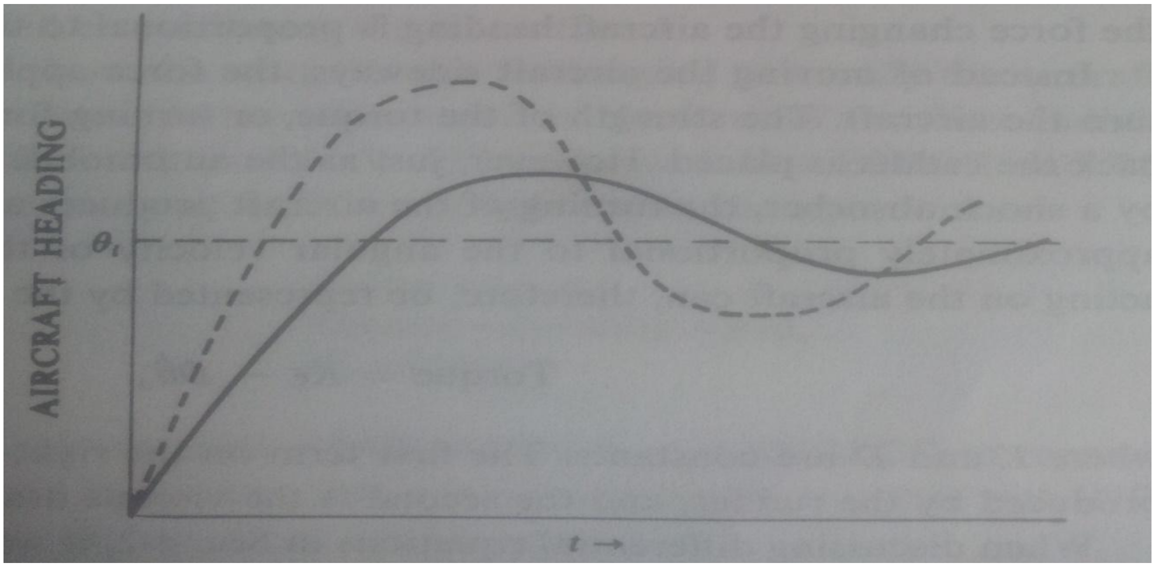


Fig: Aircraft response to autopilot system

2.6. Discrete System Simulation

In discrete systems the changes in the systems state are discontinuous. Each change in the state of the system is called an **event**. For example, the arrival or departure of a customer in a queue is an event. That contrasts to continuous systems in which the state changes smoothly with time. Simulation is the representation or imitation of a system by a computer for the purpose of predicting the behaviour of that system under certain conditions.

Discrete-event simulation is commonly used in research to study large, complex systems which are not suitable for conventional analytic approach. Some examples are the study of sea and airports, & telephone exchanges. Another use of simulation is for tuning a design, an example is network design.

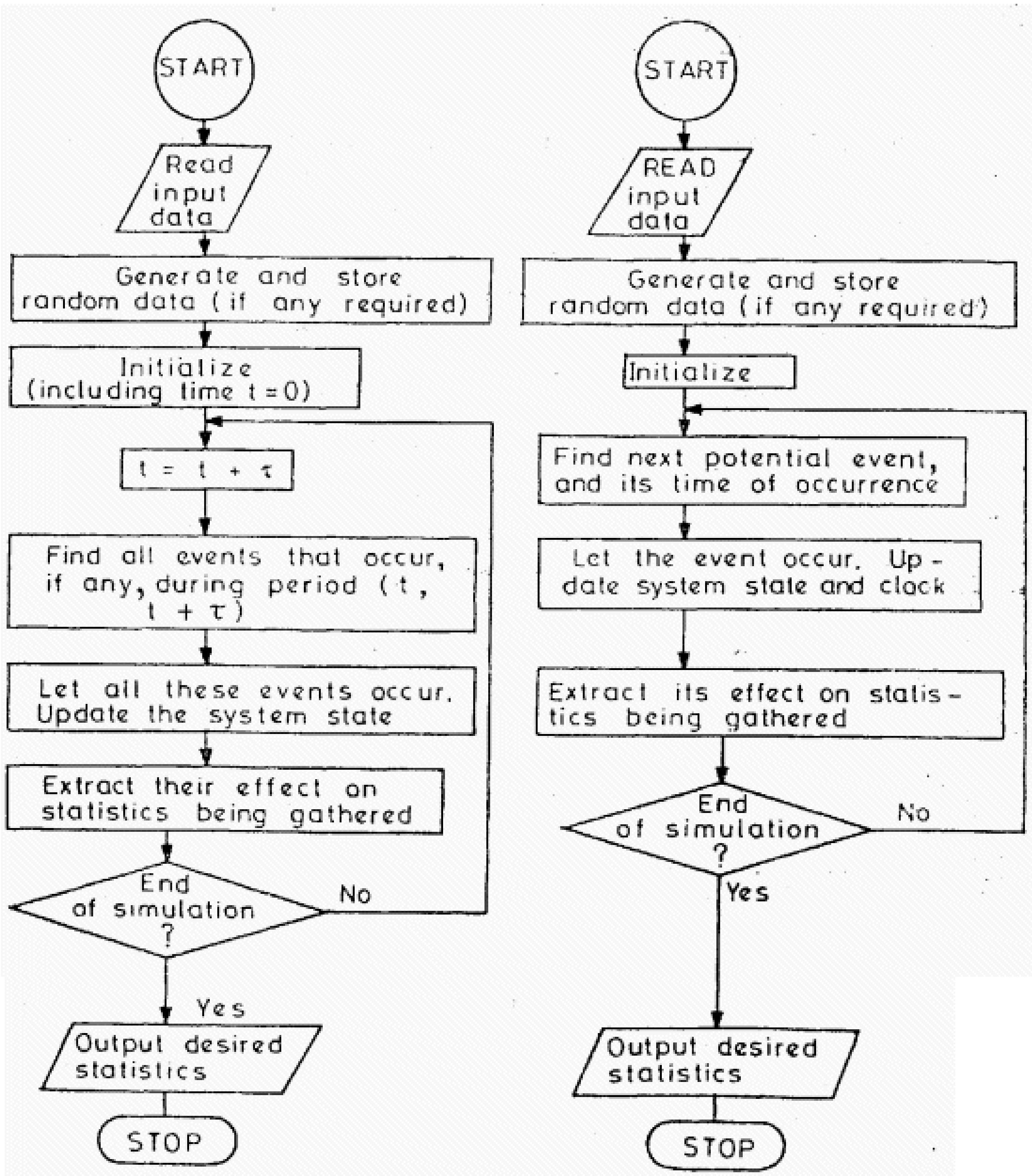
2.6.1. Representation of time in Discrete Event Simulation

Fixed time-step versus next-event model:

In a fixed time-step (**interval Oriented)** model a timer is simulated by the computer, this timer is updated by a fixed time interval τ, and the system is examined to see if any event has taken place during this time interval, all events that take place during that interval are treated as if they occurred simultaneously at the end of this interval.

In a next-event (**or event-to-event**) model the computer advances time to the occurrence of the next event, thus it shifts from one event to the next; the system state does not change in between. When something of interest happens to the system, the current time is kept track of.

The flowcharts for both models are shown below.



(a) Fixed time-step simulation (b) Next-event simulation

2.7. Arrival Process or patterns

Any queuing system must work on something − customers, parts, patients, orders, etc. We generally called them as entities or customers. Before entities can be processed or subjected to waiting, they must first enter the system. Depending on the environment, entities can arrive smoothly or in an unpredictable fashion. They can arrive one at a time or in clumps (e.g., bus loads or batches). They can arrive independently or according to some kind of correlation. A special arrival process, which is highly useful for modeling purposes, is the Markov arrival process. Both of these names refer to the situation where entities arrive one at a time and the times between arrivals are exponential random variables. This type of arrival process is memory-less, which means that the likelihood of an arrival within the next t minutes is the same no matter how long it has been since the last arrival. Examples are phone calls arriving at an exchange, customers arriving at a fast food restaurant, hits on a web site, and many others.

2.7.1. Poison Process (Stationary)

Consider random events such as the arrival of jobs at a job shop, the arrival of e-mail to a mail server, the arrival of boats to a dock, the arrival of calls to a call center, the breakdown of machines in a large factory, and so on. These events may be described by a counting function $N(t)$ defined for all $t \geq 0$. This counting function will represent the number of events that occurred in $[0, t]$. Time zero is the point at which the observation began, regardless of whether an arrival occurred at that instant. For each interval $[0, t]$, the value $N(t)$ is an observation of a random variable where the only possible values that can be assumed by $N(t)$ are the integers 0, 1, 2, ...

The counting process, $\{N(t), t \geq 0\}$, is said to be a Poisson process with mean rate λ if the following assumptions are fulfilled:

a. Arrivals occur one at a time.
b. $\{N(t), t \geq 0\}$ has stationary increments: The distribution of the number of arrivals between t and $t + s$ depends only on the length of the interval s, not on the starting point t. Thus, arrivals are completely at random without rush or slack periods.
c. $\{N(t), t \geq 0\}$ has independent increments: The number of arrivals during nonoverlapping time intervals are independent random variables. Thus, a large or small number of arrivals in one time interval has no effect on the number of arrivals in subsequent time intervals. Future arrivals occur completely at random, independent of the number of arrivals in past time intervals.

This is equivalent to saying that the number of arrivals per unit time is a random variable with a Poisson's distribution. This distribution is used when chances of occurrence of an event out of a large sample is small.

That is if X = number of arrivals per unit time, then, probability distribution function of arrival is given as

$$f(x) = \Pr(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}, \begin{cases} x = 0, 1, 2, \ldots \\ \lambda > 0 \end{cases}$$

$$E(X) = \lambda$$

Where λ. is the average number of arrivals per unit time (1/ τ), E(X) is the expected number, and x is the number of customers per unit time. This pattern of arrival is called Poisson's arrival pattern. τ is inter arrival time.

Illustrative example

In a single pump service station, vehicles arrive for fueling with an average of 5 minutes between arrivals. If an hour is taken as unit of time, cars arrive according to Poison's process with an average of

λ= 12 cars/hr.

The distribution of the number of arrivals per hour is,

$$f(x) = \Pr(X = x) = \frac{e^{-\lambda}\lambda^x}{x!} = \frac{e^{-12}12^x}{x!}, \begin{cases} x = 0, 1, 2, .. \\ \lambda > 0 \end{cases}$$

Illustrative example

In a single pump service station, vehicles arrive for fueling with an average of 5 minutes between arrivals. If an hour is taken as unit of time, cars arrive according to Poison's process with an average of

λ= 12 cars/hr.

The distribution of the number of arrivals per hour is,

$$f(x) = \Pr(X = x) = \frac{e^{-\lambda}\lambda^x}{x!} = \frac{e^{-12}12^x}{x!}, \begin{cases} x = 0, 1, 2, .. \\ \lambda > 0 \end{cases}$$

2.7.2Non-stationary Poisson Process

Non-stationary Poisson process (NSPP), is characterized by $\lambda(t)$, the arrival rate at time t. The NSPP is useful for situations in which the arrival rate varies during the period of interest, including meal times for restaurants, phone calls during business hours, and orders for pizza delivery around 6 P. M. The key to working with an NSPP is the expected number of arrivals by time t, denoted by:

$$\Lambda(t) = \int_0^t \lambda(s)\, ds$$

To be useful as an arrival-rate function, $\lambda(t)$ must be nonnegative and integrable. For a stationary Poisson process with rate λ we have $\Lambda(t) = \lambda t$, as expected.

Let $T_1, T_2, \ldots$ be the arrival times of stationary Poisson process $N(t)$ with $\lambda = 1$, and let $\mathcal{T}_1, \mathcal{T}_2, \ldots$ be the arrival times for an NSPP $\mathcal{N}(t)$ with arrival rate $\lambda(t)$. The fundamental relationship for working with NSPPs is the following:

$$T_i = \Lambda(\mathcal{T}_i)$$
$$\mathcal{T}_i = \Lambda^{-1}(T_i)$$

In other words, an NSPP can be transformed into a stationary Poisson process with arrival rate 1, and a stationary Poisson process with arrival rate 1 can be transformed into an NSPP with rate $\lambda(t)$, and the transformation in both cases is related to $\Lambda(t)$.

Example

Suppose that arrivals to a post office occur at a rate of 2 per minute from 8 A.M. until 12 P.M., then drop to 1 every 2 minutes until the day ends at 4 P.M. What is the probability distribution of the number of arrivals between 11 A.M. and 2 P.M?

Let time $t = 0$ correspond to 8 A.M. Then this situation could be modeled as an NSPP $\mathcal{N}(t)$ with rate function

$$\lambda(t) = \begin{cases} 2, & 0 \le t < 4 \\ \frac{1}{2}, & 4 \le t \le 8 \end{cases}$$

The expected number of arrivals by time t is therefore

$$\Lambda(t) = \begin{cases} 2t, & 0 \le t < 4 \\ \frac{t}{2} + 6, & 4 \le t \le 8 \end{cases}$$

Notice that computing the expected number of arrivals for $4 \le t \le 8$ requires that the integration be done in two parts:

$$\Lambda(t) = \int_0^t \lambda(s)\, ds = \int_0^4 2\, ds + \int_4^t \frac{1}{2}\, ds = \frac{t}{2} + 6$$

Since 2 P.M. and 11 A.M. correspond to times 6 and 3, respectively, we have

$$\begin{aligned} P[\mathcal{N}(6) - \mathcal{N}(3) = k] &= P[N(\Lambda(6)) - N(\Lambda(3)) = k] \\ &= P[N(9) - N(6) = k] \\ &= \frac{e^{9-6}(9-6)^k}{k!} \\ &= \frac{e^3(3)^k}{k!} \end{aligned}$$

where $N(t)$ is a stationary Poisson process with arrival rate 1.

2.7.3. Batch/Bulk Arrivals

If arriving customers to a queue occur in "batches" such as busloads, then we can model this by a point process $\psi = \{t_n\}$ in which the arrival times of customers can coincide: $t_0 \leq t_1 \leq t_2 \leq \cdots$, where $\lim_{n\to\infty} t_n = \infty$. Since the limit is infinite, we conclude that the inequalities must consist of an infinite number of strict inequalities with a finite number of equalities in between. For example $0 = t_0 = t_1 = t_2 < 1 = t_3 = t_4 = t_5 = t_6 < 3 = t_7 = t_8 < t_9 \cdots$ means that a batch of size 3 occured at the origin, followed by a batch of size 4 at time $t = 1$ followed by a batch of size 2 at time $t = 3$, and so on. If we randomly select an integer j, then C_j (the j^{th} customer; arrival time t_j) is a member within some batch. As is underlying the so-called *inspection paradox*, we are more likley to choose someone from a larger batch since larger batches contain more customers. The size of this batch is thus biased to be larger than usual.

2.8. Gathering Statistics

Most simulation programming systems include a report generator to print out statistics gathered during the run. The exact statistics required from a model depend upon the study being performed, but there are certain commonly required statistics which are usually included in the output. Among the commonly needed statistics are:

(a) *Counts* giving the number of entities of a particular type or the number of times some event occurred.

(b) *Summary measures*, such as extreme values, mean values, and standard deviations.

(c) *Utilization*, defined as the fraction (or percentage) of time some entity is engaged.

(d) *Occupancy*, defined as the fraction (or percentage) of a group of entities in use on the average.

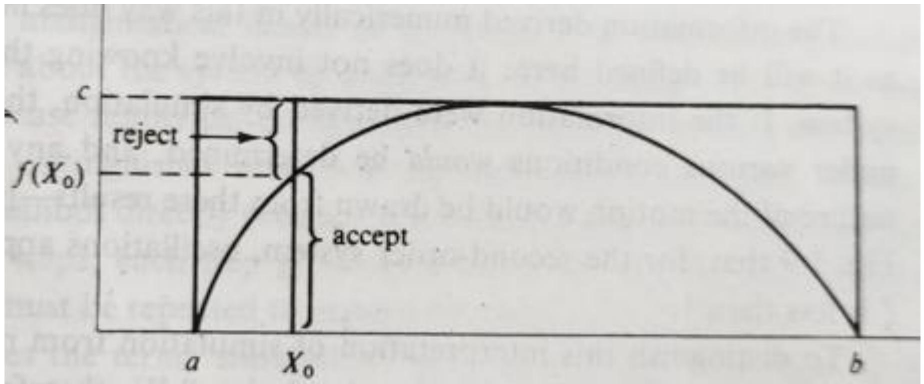
(e) *Distributions* of important variables, such as queue lengths or waiting times.

(f) *Transit times*, defined as the time taken for an entity to move from one part of the system to some other part.

When there are stochastic effects operating in the system, all these system measures will fluctuate as a simulation proceeds, and the particular values reached at the end of the simulation are taken as estimates of the true values they are designed to measure. Deciding upon the accuracy of the estimates is a problem that will be taken up in Chap. 14. For the time being, we discuss the methods used to derive the estimates.

2.9 Monte Carlo Method/ Simulation

It is a numerical computation method that consists of extensive experimental sampling with random numbers. For Example, the integral of a single variable over a given range corresponds to finding the area under the graph representing the function. Suppose the function, ***f(x)*** is positive and has lower and upper bounds ***a*** and ***b***, respectively. Suppose, also, the function is bounded above by the value ***c***.



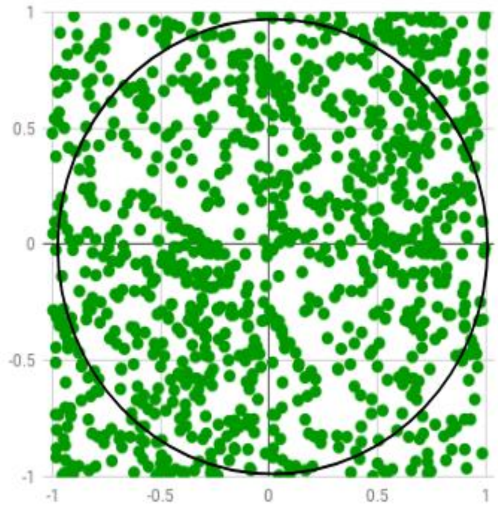
As shown in the figure above, the graph of the function is then contained within a rectangle with sides of length ***c***, and ***b-a.*** If we pick points at random within the rectangle, and determine whether they lie beneath the curve or not, it is apparent that, providing the distribution of selected points is uniformly spread over the rectangle. The fraction of the points falling on or below the curve should be approximately the ratio of the area under the curve to the area of the rectangle. If ***N*** points are used and ***n*** of them fall under the curve, then, approximately,

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

The accuracy improves as the number of ***N*** increases. When it is decided that the sufficient poits has been taken, the value of integral is estimated by multiying ***n/N*** with the area of rectangle, ***c(b-a).***

Estimating the value of Pi using Monte Carlo

The idea is to simulate random (x, y) points in a 2-D plane with domain as a square of side 1 unit. Imagine a circle inside the same domain with same diameter and inscribed into the square. We then calculate the ratio of number points that lied inside the circle and total number of generated points. Refer to the image below:



We know that area of the square is 1 unit sq while that of circle is $\pi * (\frac{1}{2})^2 = \frac{\pi}{4}$. Now for a very large number of generated points,

$$\frac{\text{area of the circle}}{\text{area of the square}} = \frac{\text{no. of points generated inside the circle}}{\text{total no. of points generated or no. of points generated inside the square}}$$

that is,

$$\pi = 4 * \frac{\text{no. of points generated inside the circle}}{\text{no. of points generated inside the square}}$$

The Algorithm

1. Initialize circle_points, square_points and interval to 0.
2. Generate random point x.
3. Generate random point y.
4. Calculate d = x*x + y*y.
5. If d <= 1, increment circle_points.
6. Increment square_points.
7. Increment interval.
8. If increment < NO_OF_ITERATIONS, repeat from 2.
9. Calculate pi = 4*(circle_points/square_points).
10. Terminate.