Unit-2: Simulation of Continuous and Discrete System

Continuous System Models:

Many systems comprise computers, communications networks, and other digital systems to monitor and control physical (electrical, mechanical, thermodynamic, etc.) processes. Models of these systems have some parts modeled as discrete event systems, other parts modeled with continuous (differential or differential-algebraic) equations, and the interaction of these parts is crucial to understanding the system's behavior.

Following points describes the concept of continuous system:

- In *continuous simulation*, continuously changing state variables of a system are modeled by differential equations.
- A continuous system is the system in which the activities of the main elements of the system cause smooth changes in the attributes of the entities of the system.
- On mathematical modeling, the attributes of the system are controlled by the continuous functions.
- In such system, the relationships depicts the rates at which the attributes changes.
- The continuous system is modeled using the differential equations.
- The complex continuous system with non-linearity can be simulated by showing the application to models for linear differential equations to obtain constant coefficients and then generalize to more complex equations.

Analog Computer:

Analog computers are those computers that are unified with devices like Integrator, Differentiator, Adder, Inverter, Multiplier, and Divider so as to simulate the continuous mathematical model of the system, which generates continuous outputs.

Analog method of system simulation is for use of analog computer and other analog devices in the simulation of continuous system. The analog computation is sometimes called differential analyzer.

PRATIK CHAND

The most widely used form of analog computer is the electronic analog computer, based on the use of high gain direct current (dc) amplifiers called operational amplifiers. Voltages in the computer are equated to mathematical variables and the operational amplifiers can add and integrate the voltages.

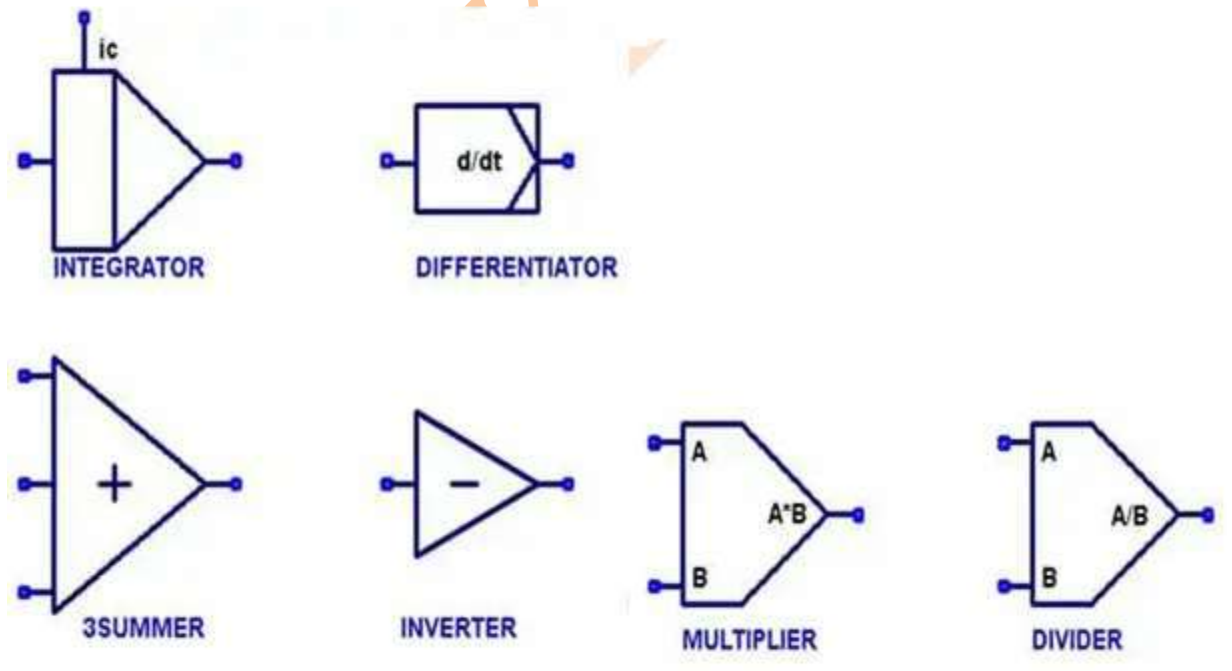
Electronic analog computers are limited in accuracy. It is difficult to carry the accuracy of measuring a voltage beyond a certain point. Secondly, many assumptions are made. Also, operational amplifiers have a limited dynamic range of output.

But many users prefer to use analog computers. The analog representation of a system is often more natural as it directly reflects the structure of the system thus simplifying both the setting-up of a simulation and the interpretation of results. Also analog computer can be solving many equations in a truly simultaneous manner so is faster.

CHAND

Analog Methods:

Following are the analog method or symbols used to design the analog computer.



Example:

The general method by which analog computers are applied can be demonstrated using second order differential equation.

$M\ddot{x} + D\dot{x} + kx = kF(t)$

Solving the equation for the highest order derivate gives,

$M\ddot{x} = kF(t) - D\dot{x} - kx$

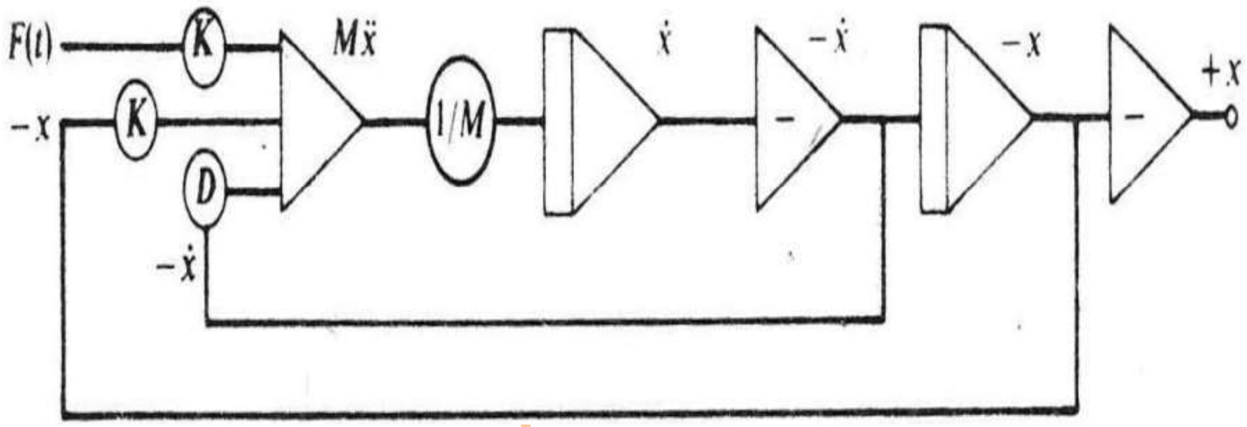


Fig: Analog simulation of automobile suspension system

Suppose a variable representing the input (t) is supplied, assume there exists variables representing $-x$ and $-\dot{x}$. These three variables can be scaled and added to produce $M\ddot{x}$. Integrating it with a scale factor $1/M$produces $\dot{x}$. Changing sign produces $-\dot{x}$, further integrating produces $-x$, a further sign inverter is included to produce $+x$ as output.

When a model has more than one independent variable, a separate block diagram is drawn for each independent variable and where necessary, interconnections are made between the diagrams.

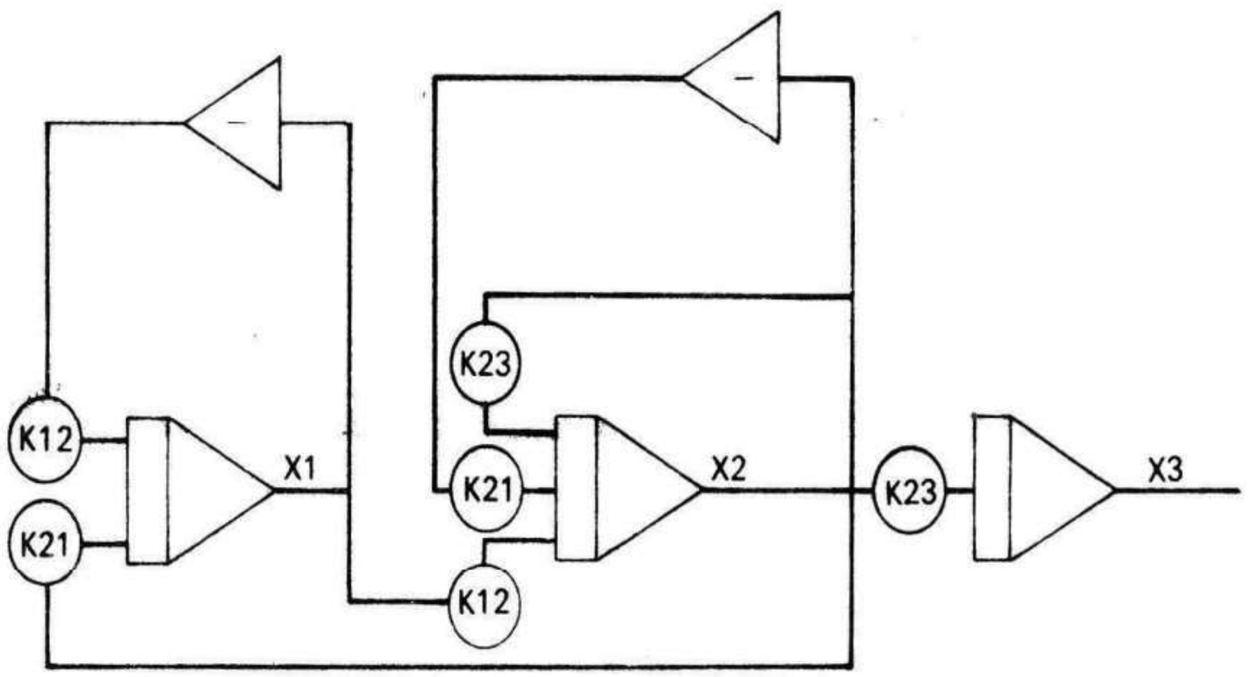
Example: Design analog computer of

$x'_1 = -k12x1 + k21x2$

$x'_2 = k12x1 - (k21 \text{ - } k23)\ x2$

$x'_3 = k23x2$

PRATIK CHAND



PRATIK CH

There are three integrators. Reading from left to right, they solve the equations for $x1$, $x2$ & $x3$. Interconnections between the three integrators with sign changers where necessary provides inputs that define the differential coefficients of the three variables.

First integrator, for example is solving the equation,

$x'_1 = -k12x1 + k21x2$

The second integrator is solving the equation

$x'_2 = k12x1 - (k21 - k23)2$

$= k12x1 - k21x2 + k23x2$

In this case, the variable $x2$ is being used twice as an input to the integrator, so that the two coefficients $k21$ and $k23$ can be changed independently. The last integrator solves the equation

$x'_3 = k23x2$

Assignment:

Q1) Design analog computer of

$x1^{\cdot\cdot} - kx1^{\cdot} + k1x1^{\cdot\cdot\cdot} = a\ x3$

$x2^{\cdot\cdot\cdot} + k2x1 - k1\ x3 - k3x2 = 0$

$x3^{\cdot\cdot\cdot} - k2x2 + x1^{\cdot\cdot} = f(x)$

Q2) Design analog computer of

$y1^{\cdot} - x1^{\cdot} + k1\ z1^{\cdot\cdot\cdot} = 0$

$x1^{\cdot\cdot} = x1 - x1^{\cdot}z1$

$z1^{\cdot\cdot\cdot} = k2x1 + x1^{\cdot\cdot\cdot}$

Q3) Design analog computer of

$x^{\cdot} - kx^{\cdot} + kx^{\cdot\cdot\cdot} = x$

$y^{\cdot} = x - k\ z - k3x$

$z^{\cdot\cdot\cdot} = k\ y + z^{\cdot}$

PRATIK CHAND

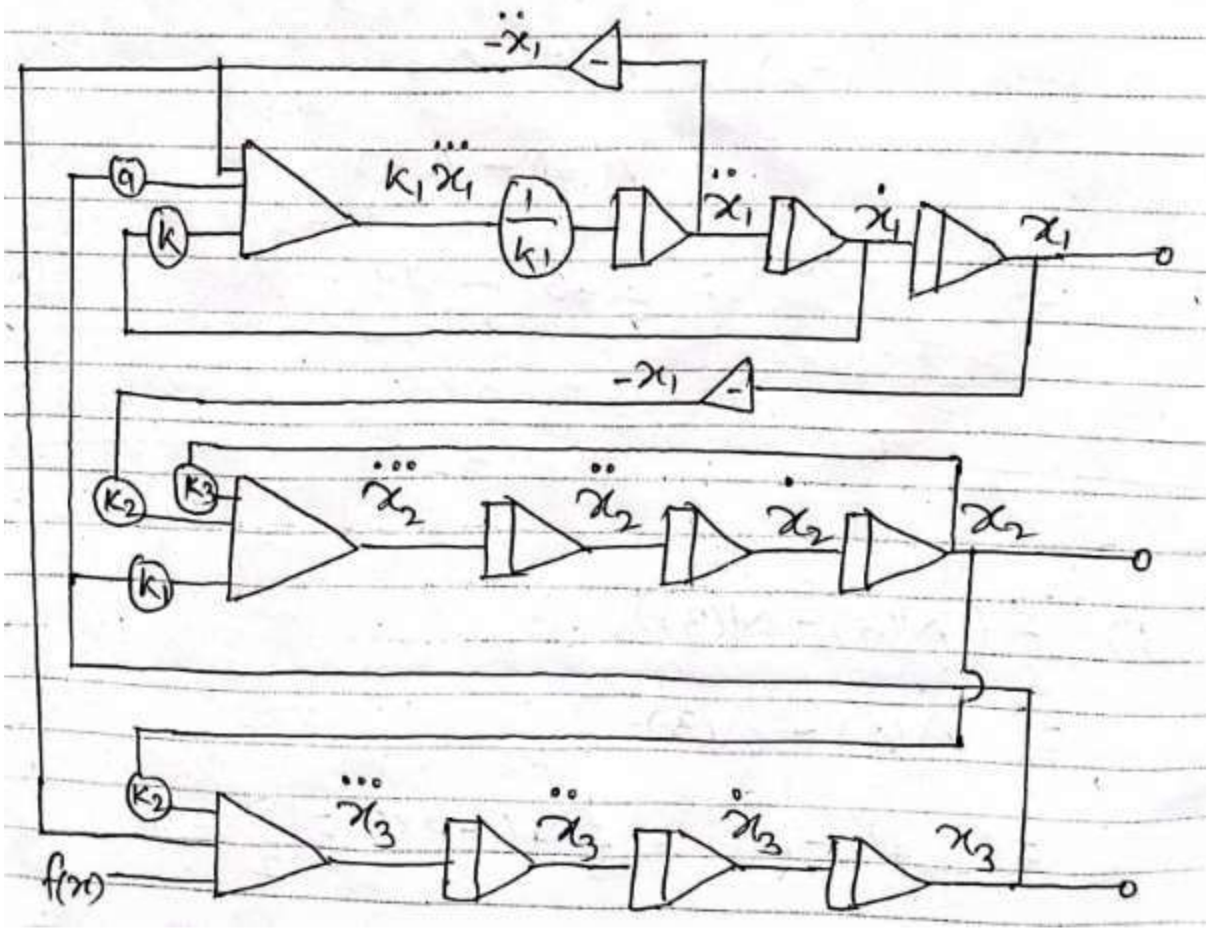
Solution of Assignment- Analog Computer Design

Q1) Solution:-

$$k_1 \dddot{x}_1 = a x_3 - \ddot{x}_1 + k \dot{x}_1$$

$$\dddot{x}_2 = k_1 x_3 + k_3 x_2 - k_2 x_1$$

$$\dddot{x}_3 = f(x) + k_2 x_2 - \ddot{x}_1$$

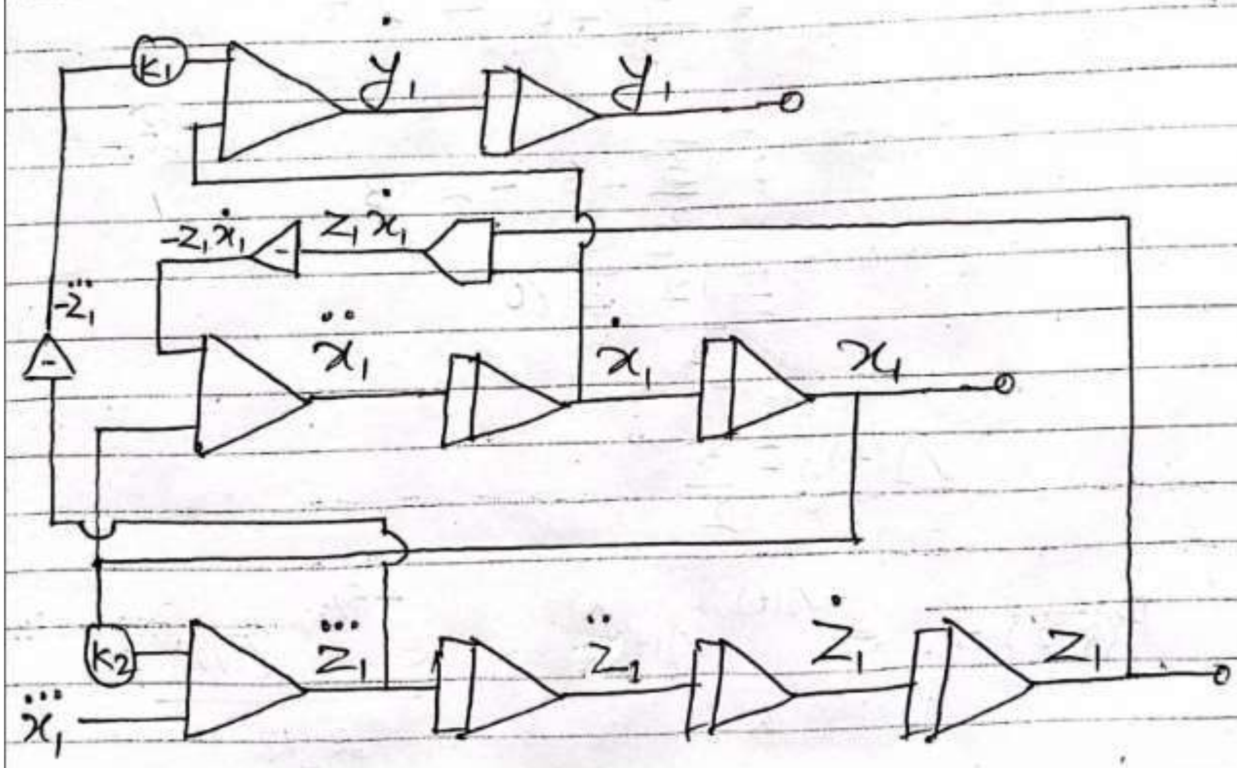


Q2) Solution

$$\dot{y}_1 = \dot{x}_1 - k_1 \dddot{z}_1$$

$$\ddot{x}_1 = x_1 - \dot{x}_1 z_1$$

$$\dddot{z}_1 = k_2 x_1 + \dddot{x}_1$$

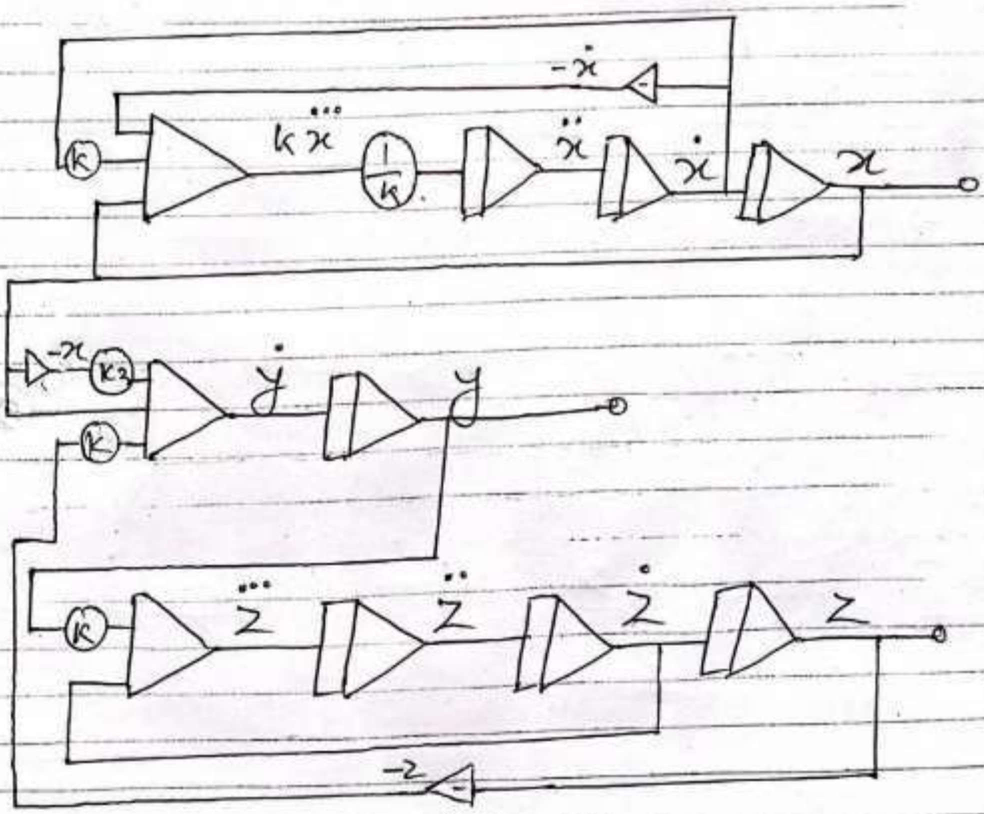


Q3) Solution

$$k\dddot{x} = x - \dot{x} + k\dot{x}$$

$$\dot{y} = x - kz - k_3 x$$

$$\dddot{z} = ky + \dot{z}$$



Hybrid Simulation:

For most studies the model is clearly either of a continuous or discrete nature and that is the determining factor in deciding whether to use an analog or digital computer for system simulation. However, there are times when an analog and digital computers are combined to provide simulation. In this circumstances hybrid simulation is used. Hybrid simulation is provided by combining analog and digital computers. The form taken by hybrid simulation depends upon the applications.

Here, one computer may be simulating the system being studied while other is providing a simulation of the environment in which the system is to operate. It is also possible that the system being simulated is an interconnection of continuous and discrete subsystems, which can be modeled by an analog and digital computer being linked together.

The introduction of hybrid simulation required certain technological developments for its exploitation. High speed converters are needed to transform signals from one form of representation to the other.

Practically, the availability of mini computers has made hybrid simulation easier, by lowering costs and allowing computers to be dedicated to an application. The term "hybrid simulation" is generally reserved for the case in which functionality distinct analog and digital computers are linked together for the purpose of simulation.

Digital-Analog Simulators:

- To avoid the disadvantages of analog computers, many digital computer programming languages have been written to produce digital analog simulators.
- They allow a continuous model to be programmed on a digital computer in essentially the same way as it is solved on an analog computer.
- The language contains macro instructions that carryout the actions of adders, integrators and sign changers.
- More powerful techniques of applying digital computers to the simulation of continuous systems have been developed.
- As a result, digital analog simulators are not now in expensive use.

PRATIK CHAND

Interactive System

Interactive systems are computer systems characterized by significant amounts of interaction between humans and the computer. Editors, CAD-CAM (Computer Aided Design-Computer Aided Manufacture) systems, and data entry systems are all computer systems involving a high degree of human-computer interaction. Games and simulations are interactive systems. Web browsers and Integrated Development Environments (IDEs) are also examples of very complex interactive systems.

Feedback System:

The system takes feedback from the output i.e. input is coupled with output. A significant factor in the performance of many systems is that coupling occurs between the input and output of the system. The term feedback is used to describe the phenomenon.

There are two types of feedback system:

- Negative feedback system
- Positive feedback system

Negative feedback system:

A negative feedback system is a regulatory mechanism in which the output or response of a process opposes the initial change that triggered it. In this system, when a particular event or signal occurs, it leads to a response that counteracts or reduces the intensity or magnitude of that event, restoring the system to its original state or equilibrium.

In simple terms, negative feedback "feeds back" less of the output into the system, which helps to stabilize and control the process. It plays a crucial role in maintaining homeostasis and stability in various natural and artificial systems.

Negative feedback loops are commonly found in biological systems, such as the regulation of body temperature, blood pressure, and hormone levels. In these cases, when a certain parameter deviates from its set point, negative feedback mechanisms kick in to bring it back within the desired range.

PRATIK CHAND

Additionally, negative feedback is widely used in engineering and control systems to ensure stability and accuracy. By continuously monitoring the output and adjusting it in the opposite direction to the deviation, negative feedback helps maintain the desired operating conditions.

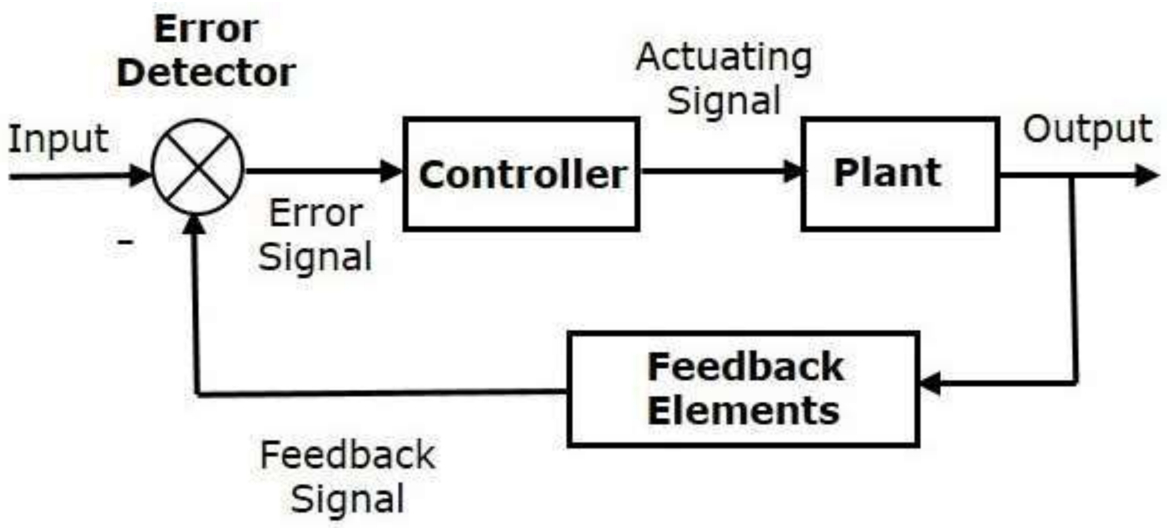


Fig: Negative feedback system

Positive feedback system:

A positive feedback system is a type of regulatory mechanism in which the output or response of a process amplifies or reinforces the initial change that triggered it. In this system, when a particular event or signal occurs, it leads to an increase in the intensity or magnitude of that event, resulting in further stimulation of the process.

In simple terms, positive feedback "feeds back" more of the output into the system, promoting its continuation and amplification. This can lead to exponential growth or escalation in certain situations. Positive feedback loops can be found in various natural and artificial systems, including biology, economics, climate, and engineering.

It's important to note that while positive feedback can lead to dramatic changes, it can also make systems more unstable and prone to reaching extremes. As a result, some systems have built-in mechanisms to counteract or dampen positive feedback loops to maintain stability. On the other hand, negative feedback systems work in

PRATIK C

the opposite way, where the response opposes the initial change and helps to maintain equilibrium or balance in the system.

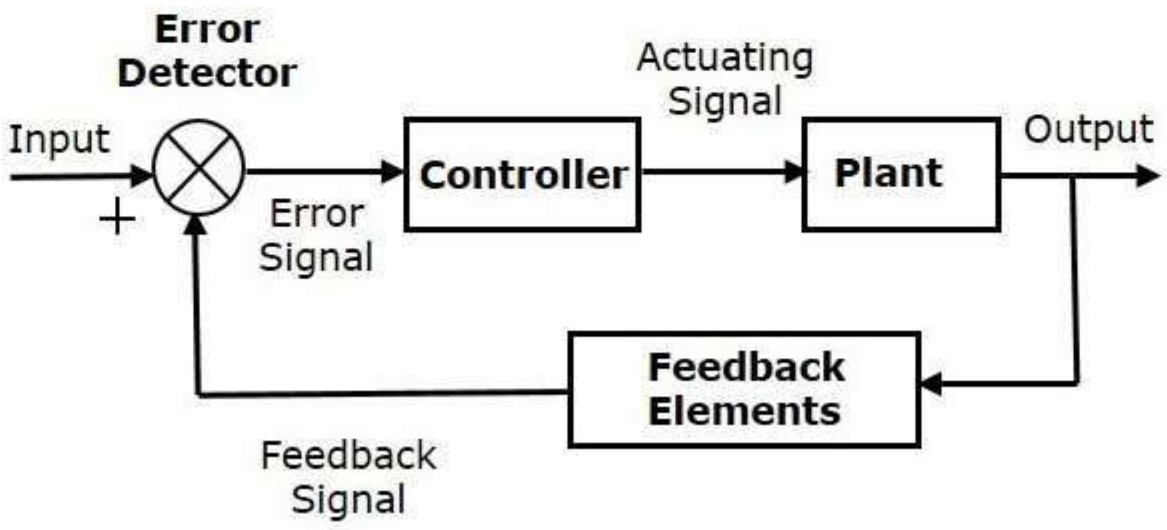


Fig: Positive feedback system

Example;

- Autopilot aircraft system
- A home heating system controlled by a thermostat
- Error Correction mechanism

Autopilot aircraft System:

In the aircraft feedback system, 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 (desired and actual output). 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 control the heading.

The difference between the desired signal $\boldsymbol{\theta_t}$ and actual heading $\boldsymbol{\theta_0}$ is called the error signal, since it is a measure of the extent to which the system from the desired condition. It is denoted by ϵ.

PRATIK CH

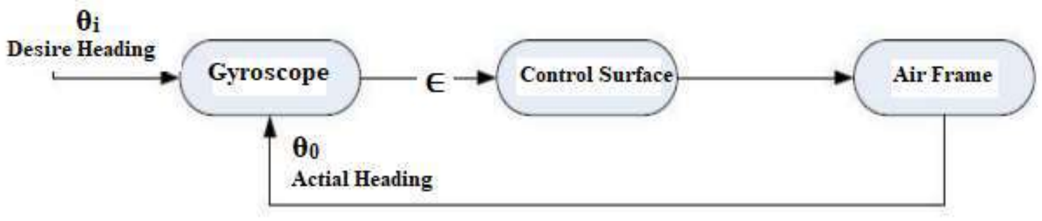


Fig: An aircraft under autopilot control (Continuous system)

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, $\boldsymbol{\theta_i}$, and the actual heading, $\boldsymbol{\theta_0}$. 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)

PRATIK CHAND

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

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

This is a second order differential equation.

Note: Torque is the measure of the force that can cause an object to rotate about an axis.

Graphical representation of autopilot aircraft is shown in figure bellow:

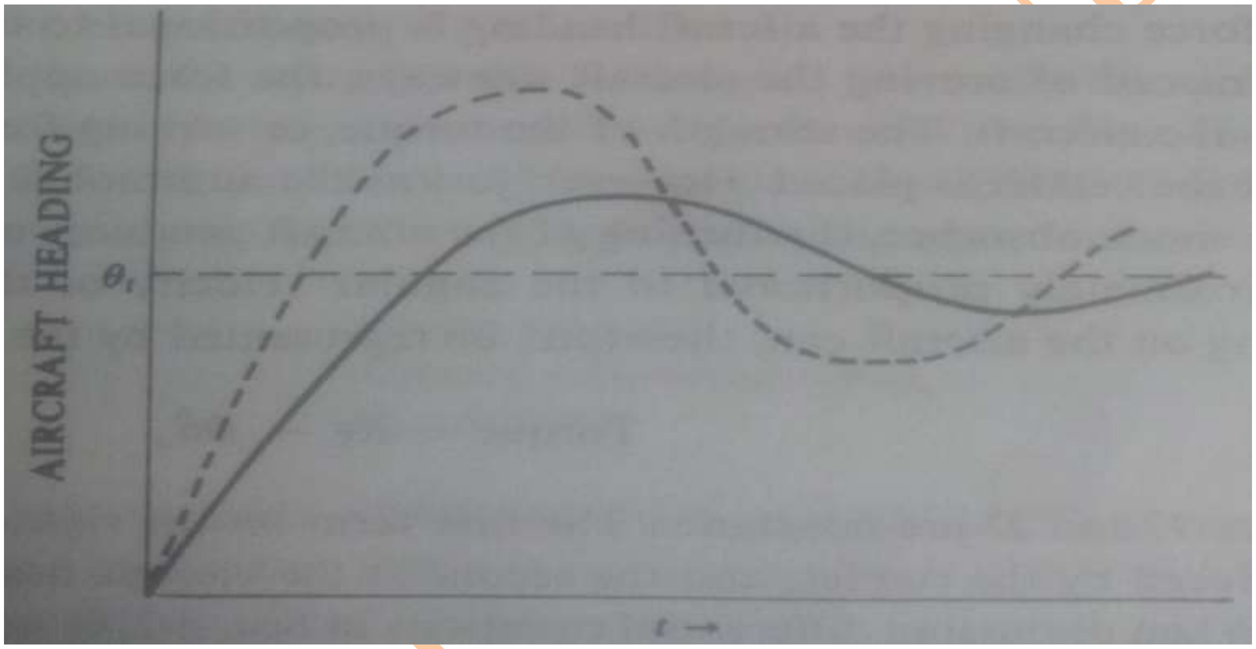


Fig: Aircraft response to autopilot system

Home heating system:

The system has a furnace whose purpose is to heat a room, and the output of the system can be measured as a 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 feedback from the output to input. In this case there are only two states either the furnace is on or off.

Discrete-Event Simulation:

Discrete-event simulation, or DES, is intended to simulate systems where events occur at specific, separable instances in time. DES can be either deterministic or stochastic, depending on the nature of the target process.

Each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation time can directly jump to the occurrence time of the next event, which is called **next-event time progression**.

In addition to next-event time progression, there is also an alternative approach, called **incremental time progression**, where time is broken up into small time slices and the system state is updated according to the set of events/activities happening in the time slice. Because not every time slice has to be simulated, a next-event time simulation can typically run faster than a corresponding incremental time simulation.

Both forms of DES contrast with continuous simulation in which the system state is changed continuously over time on the basis of a set of differential equations defining the rates of change of state variables.

Example:

A common exercise in learning how to build discrete-event simulations is to model a Queuing, such as customers arriving at a bank teller to be served by a clerk (*bank staff*).

In this example, the system objects are ***Customer*** and ***Teller***, while the system events are ***Customer-Arrival***, ***Service-Start*** and ***Service-End***. Each of these events comes with its own dynamics defined by the following event routines:

1. When a ***Customer-Arrival*** event occurs, the state variable ***queue-length*** is incremented by 1, and if the state variable ***teller-status*** has the value "**available**", a ***Service-Start*** follow-up event is scheduled to happen without any delay, such that the newly arrived customer will be served immediately.
2. When a ***Service-Start*** event occurs, the state variable ***teller-status*** is set to "**busy**" and a ***Service-End*** follow-up event is scheduled with a delay (obtained from sampling a ***service-time*** random variable).

PRATIK CHAND

3. When a ***Service-End*** event occurs, the state variable ***queue-length*** is decremented by 1 (representing the customer's departure). If the state variable ***queue-length*** is still greater than zero, a ***Service-Start*** follow-up event is scheduled to happen without any delay. Otherwise, the state variable ***teller-status*** is set to "**available**".

The random variables that need to be characterized to model this system stochastically are the ***inter-arrival-time*** for recurrent ***Customer-Arrival*** events and the ***service-time*** for the delays of ***Service-End*** events.

The attributes of discrete-event simulation:

At a high-level, discrete-event simulation is built on top of the following components:

- **System** – a collection of entities with certain attributes.
- **State** – a collection of attributes representing the system's entities.
- **Event** – an occurrence in time that may alter the system's state.

Components of Discrete-Event Simulation:

- **System state** – a collection of variables that represent the state of the simulated system.
- **Simulation clock** – a variable that tracks the passage of time in the simulated system.
- **Event list** – a collection of when each type of event will occur.
- **Statistical counters** – variables that contain statistical information about the system, such as average request processing time, server load, or average queue length.
- **Initialization routine** – the routine that defines how the model selects the first event at time 0.
- **Timing routine** – the routine that selects the next event from the event list and jumps in time to its execution.
- **Report generator** – the routine that defines how the program calculates and updates statistical counters and includes them in a post-run report.
- **Event routine** – a routine that updates the system's state when a particular type of event occurs.

PRATIK CHAND

- **Library routines** – routines that use probability distributions to generate random values for uncertain variables in the system.
- **Main program** – a program that invokes the model's routines to perform the simulation.

<u>Representation of time in Discrete Event Simulation</u>

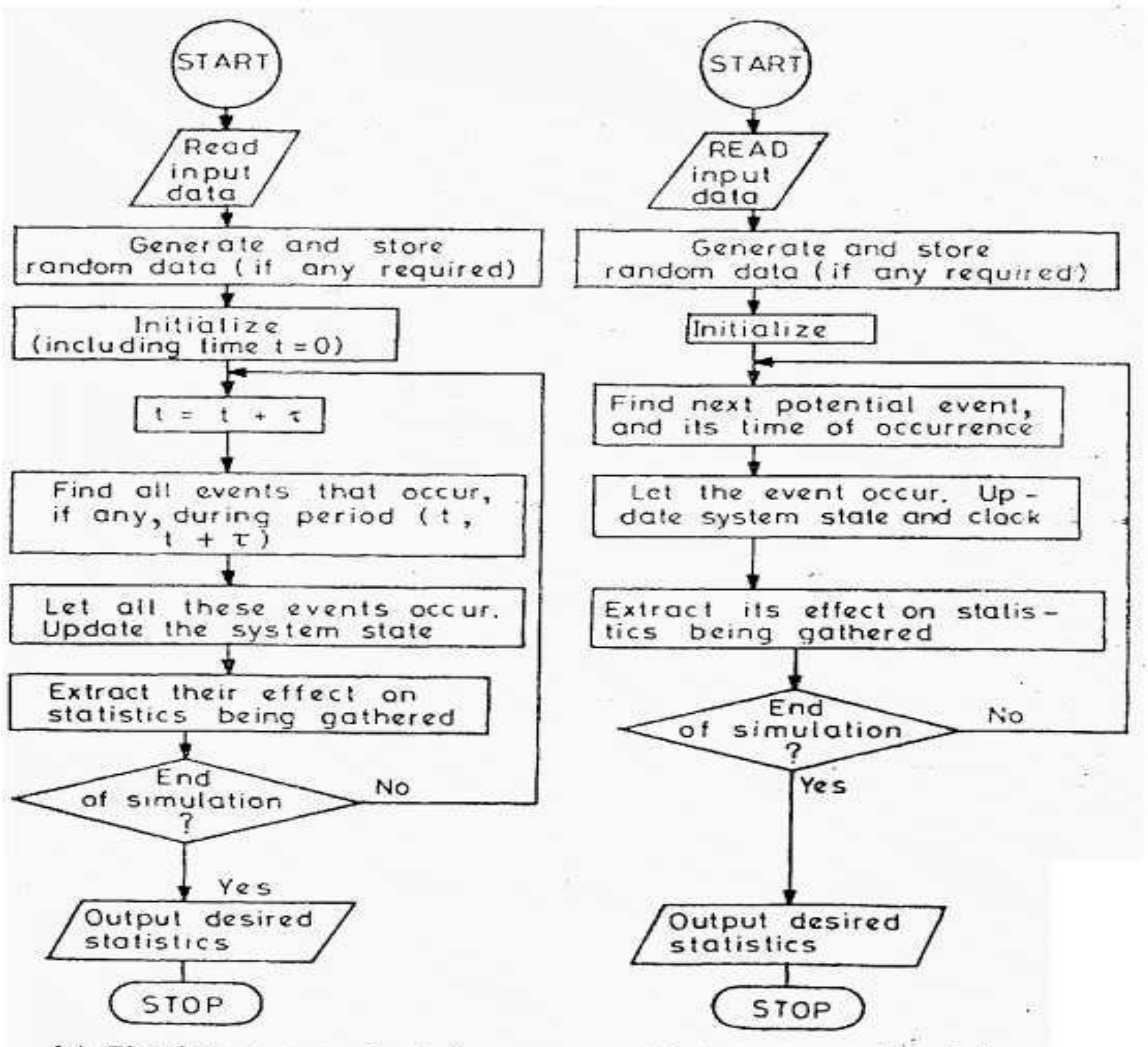
Fixed time-step versus next-event model:

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 figure below:

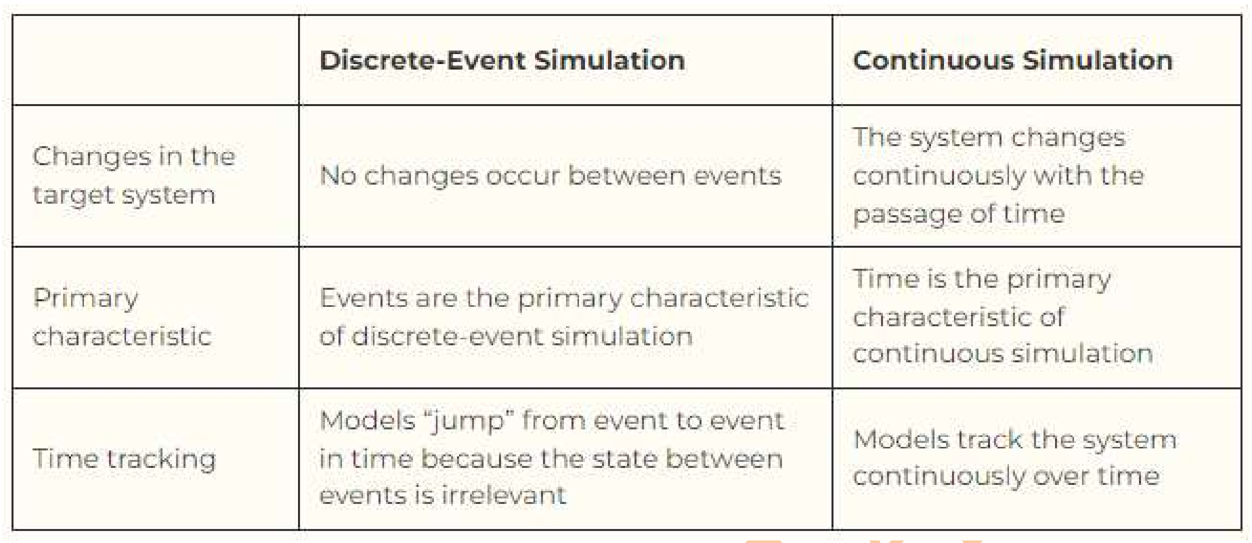
PRATIK CHAND


START
Read input data
Generate and store random data (if any required)
Initialize (including time t = 0)
t = t + τ
Find all events that occur, if any, during period (t, t + τ)
Let all these events occur. Update the system state
Extract their effect on statistics being gathered
End of simulation ?
No
Yes
Output desired statistics
STOP
START
READ input data
Generate and store random data (if any required)
Initialize
Find next potential event, and its time of occurrence
Let the event occur. Update system state and clock
Extract its effect on statistics being gathered
End of simulation ?
No
Yes
Output desired statistics
STOP

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

PRA

Comparison between DES and CS:



	Discrete-Event Simulation	Continuous Simulation
Changes in the target system	No changes occur between events	The system changes continuously with the passage of time
Primary characteristic	Events are the primary characteristic of discrete-event simulation	Time is the primary characteristic of continuous simulation
Time tracking	Models "jump" from event to event in time because the state between events is irrelevant	Models track the system continuously over time

Simulation Time and Simulation Clock:

- When you simulate a model related to time (for example, a transition with a time trigger), Model Analyst will obtain simulation time from a simulation clock.
- The simulation time is the amount of time spent on simulating a model.
- Model Analyst also uses the simulation time in a timestamp of a signal instance in the Simulation Log, in a time series chart, and on messages of a generated Sequence diagram.

There are three types of simulation clocks in Model Analyst:

- **Built-in clock**: This is the default simulation clock.
- **Internal simulation clock**: This clock is designed to precisely control the simulation time. Its implementation is based on UML run-to-completion semantics and internal completion events.
- **Model-based clock**: You can select the model-based clock by making the property as the time value tag definition of a Simulation Configuration.

PRATIK CHA

Arrival Processes:

- How customers arrive e.g. singly or in groups (batch or bulk arrivals)
- How the arrivals are distributed in time (e.g. what is the probability distribution of time between successive arrivals (the ***inter-arrival time distribution***)
- Whether there is a finite population of customers or (effectively) an infinite number

Concept of Arrival Pattern:

In simulation, the concept of "arrival pattern" refers to the pattern or distribution of entities arriving at a particular system or process over time. These entities could be customers, jobs, requests, vehicles, or any other objects that interact with the system being simulated.

Arrivals may occur at scheduled times or at random times. When at random times, the inter arrival times are usually characterized by a probability distribution and most important model for random arrival is the Poisson process. In schedule arrival inter-arrival time of customers are constant.

The arrival pattern is an essential aspect of many simulation models, as it helps to mimic real-world scenarios and analyze system behavior under different conditions.

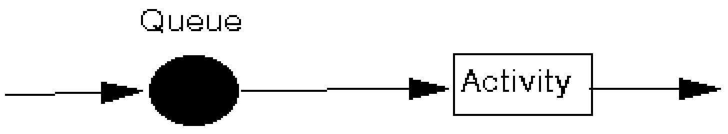


Fig: Simple queuing system

The arrival pattern can be described by various characteristics, including:

- **Arrival Rate:** The rate at which entities arrive per unit of time. It indicates how frequently new entities enter the system. For example, if a simulation is modeling a queue of customers at a store, the arrival rate could be measured in customers per minute.
- **Inter-arrival Time:** The time interval between the arrival of consecutive entities. It is the reciprocal of the arrival rate. For example, if the arrival rate

PRATIK CHAND

is 5 customers per minute, the average inter-arrival time would be 1/5 = 0.2 minutes (or 12 seconds).

- **Arrival Time Distribution:** The probability distribution that describes when entities arrive. Common distributions used in simulation include uniform, exponential, Poisson, normal, and others. Each distribution represents different patterns of arrival times.
- **Arrival Patterns over Time:** The arrival pattern may not remain constant throughout the simulation. In some cases, it might change during different time intervals, reflecting the dynamic nature of real-world systems. For example, in a call center simulation, the arrival rate might increase during peak hours and decrease during off-peak hours.

Generation of arrival pattern:

Generating arrival patterns in simulation involves creating a stream of entities arriving at a system according to a specified arrival rate and distribution. The process can be implemented using various techniques, depending on the complexity of the system and the desired characteristics of the arrival pattern.

Some common methods for generating arrival patterns in simulation:

- Stationary Poisson Process
- Non-Stationary Poisson Process
- Batch Arrival

Stationary Poisson Process:

In simulation, the Poisson process is a widely used stochastic process to model random, independent arrivals of entities into a system over time. It is particularly useful when dealing with systems where events happen independently and at an average constant rate, such as customers at a service center, packets in a computer network, or defects in a manufacturing process.

The Poisson process is characterized by its memorylessness property, which means the time until the next event does not depend on the time since the last event. This property makes it a suitable model for various real-world scenarios.

PRATIK CHAND

In this process the *average time* between events is known, but the exact timing of events is random.

The average time of event occurrence is denoted by λ (lambda),

Example 1: In a single pump service station, vehicles arrive for fueling with an average of 5 minutes between arrivals.

i.e. 1 care per 5 minutes

If an hour is taken as unit of time, cars arrive according to Poison's process with an average of

λ= 12 cars/hr.

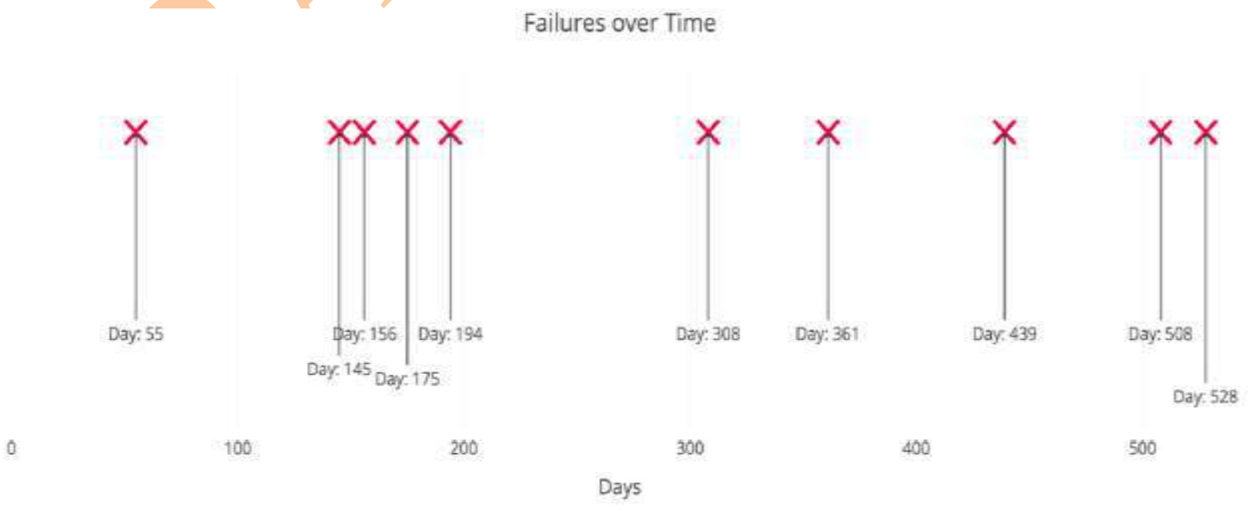
The distribution of the number of arrivals per hour is,

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

Example 2:

Suppose we own a website which our content delivery network (CDN) tells us goes down on average once per 60 days, but one failure doesn't affect the probability of the next. All we know is the average time between failures.

This is a Poisson process that looks like:



PRATIK CHAND

Here,

No of event occurs (n) = 10

Total time period (T) = 600

Average time between failure (λ) = 600/10 = 60 days

The important point is we know the *average time between events* but they are randomly spaced (stochastic). We might have back-to-back failures, but we could also go years between failures due to the randomness of the process.

A Poisson Process meets the following criteria (in reality many phenomena modeled as Poisson processes don't meet these exactly):

- Events are **independent** of each other. The occurrence of one event does not affect the probability of another event will occur.
- The average rate (events per time period) is constant.
- Two events cannot occur at the same time.

Common examples of Poisson Process are:

- Customers calling a help center
- Visitors to a website
- Movement in a stock price
- The number of hot dogs sold by Papaya King from 12pm to 4pm on Sunday
- Failures of ultrasound machines in a hospital
- The number of vehicles passing through some intersection from 8am to 11am on weekdays.

PRATIK CHAND

<u>Poisson distribution:</u>

A Poisson distribution is a discrete probability distribution. It gives the probability of an event happening a certain number of times (k) within a given interval of time or space.

"Events" could be anything like disease cases, customer purchases, failure in system, vehicle movement in road etc. The interval can be any specific amount of time or space, such as 10 days or 5 square inches.

The Poisson distribution has only one parameter, λ (lambda), which is the mean (average) number of events.

A discrete random variable **X** is said to have a Poisson distribution, with parameter **λ > 0,** if it has a probability mass function given by:

$$f(k;\lambda) = \Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

Where,

- k is the number of occurrences (k=0,1,2,3…..)
- e is Euler's number (e =2.71828)
- λ is average rate of occurrence of variable x

In Poisson distribution, the mean is represented as **E(X) = λ.** For a Poisson Distribution, the mean and the variance are equal. It means that **E(X) = V(X)**, Where, V(X) is the variance.

If percentage of occurrence is given then we can calculate λ by:

λ = np

Where,

n = total number of event

p = percentage of occurrence

You can use a Poisson distribution if:

1. Individual events happen at random and independently. That is, the probability of one event doesn't affect the probability of another event.
2. You know the mean number of events occurring within a given interval of time or space. This number is called λ (lambda), and it is assumed to be constant.

When events follow a Poisson distribution, λ is the only thing you need to know to calculate the probability of an event occurring within a certain number of times.

PRATIK CHAND

Example 1:

The number of accident in the street of Kathmandu follows Poisson distribution with mean of 3 accidents per day. Find the probability that

a. No accident occurs in a day
b. More than 3 accident in a day
c. Less than 3 accident in a day
d. Exactly 2 accidents in a day

Solution:

Given, mean (average) accident per day (λ) = 3

a. No accident (k) = 0

$$f(k;\lambda) = \Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

Where, e = 2.718 (this is constant)

$P(k=0) = (\lambda^0 e^{-3})/0!$

$= 1*(2.718)^{-3} / 1$

= 0.0498

Hence, probability of no accident occurs in a day is 0.0498.

b. More than 3 accident (k>3)

$P(k>3) = 1 – (P(k=0) + P(k=1) + P(k=2) + P(k=3))$

New,

P(k=0) = 0.0498

P(k=1) = $(\lambda^1 e^{-3})/1!$

= (3* 0.0498)/1

= 0.1494

PRATIK CHAND

P(k=2) = $(\lambda^2 e^{-3})/2!$

= $(3^2 * 0.0498)/2$

= (9 * 0.0498)/2

= 0.2241

P(k=3) = $(\lambda^3 e^{-3})/3!$

= $(3^3 * 0.0498)/6$

= (27*0.0498)/6

= 0.2241

Now

P(k>3) = 1 – (0.0498+0.1494+0.2241 +0.2241)

= 1- 0.6474

= 0.3526

Hence, probability of more than 3 accidents occurs in a day is 0.3526.

c. **Less than 3 accident (k<3)**

P(k<3) = P(k=0) + P(k=1) + P(k=2)

= 0.0498+0.1494+0.2241

= 0.4233

Hence, probability of less than 3 accidents occurs in a day is 0.4233.

d. **Exactly 2 accident in a day (k=2)**

P(k=2) = 0.2241

Hence, probability of exactly 2 accidents occurs in a day is 0.2241.

PRATIK CHAND

Example 2:

A manufacturer company of pins knows on an average 2% of its production is defective. Company sells pins in box of 100 and guarantees that not more than 2 pins will be defective in a box. What is the probability that a box selected at random:

a. Will meet the guaranteed quality
b. Will not meet the guaranteed quality

Solution:

Here given,

Average defective percentage (p) = 2% = 0.02

Total no of pins in a box (n) = 100

Average defective pins (λ) = n*p = 100*0.02 = 2

e = 2.718 (this is constant)

a. Guaranteed will meet when (k<=2)

P(k<=2) = P(k=0) + P(k=1) + P(k=2)

P(k=0) = $(\lambda^0 e^{-2})/0!$

= (1* 0.1354)/1

= 0.1354

P(k=1) = $(\lambda^1 e^{-2})/1!$

= (2* 0.1354)/1

= 0.2708

P(k=2) = $(\lambda^2 e^{-2})/2!$

= (4* 0.1354)/2

= 0.2708

PRATIK CHAND

P(k<=2) = 0.1354+0.2708+0.2708

= 0.677

Hence if box is selected randomly probability of guaranteed quality is 0.677.

b. Guaranteed will not meet when (k>2)

P(k>2) = 1- (P(k=0) + P(k=1) + P(k=2))

= 1- 0.677

= 0.323

Hence if box is selected randomly probability of not guaranteed quality is 0.323.

Example 3:

A random variable X has a Poisson distribution with parameter λ such that

P (X = 1) = (0.2) P (X = 2). Find P (X = 0).

Solution:

For the Poisson distribution, the probability function is defined as:

$P(X=k) = (e^{-\lambda} \lambda^k) / k!$, where λ is a parameter.

Given that, P (X = 1) = (0.2) P (X = 2)

X = k, then

P (k = 1) = (0.2) P (k = 2)

$(e^{-\lambda} \lambda^1) / 1! = (0.2) (e^{-\lambda} \lambda^2) / 2!$

$e^{-\lambda} \lambda = (0.2) (e^{-\lambda} \lambda^2) / 2$

$\lambda = 2/0.2$

$\lambda = 10$

Now, substitute $\lambda = 10$, in the formula, we get:

P (X =0),

PRATIK CHAND

X=k then,

P (k =0) = $(e^{-\lambda} \lambda^0)/0!$

= $e^{-\lambda}$ (λ^0 = 1 and 0! = 1)

= e^{-10}

= 0.0000454

Thus, probability of zero occurrences is 0.0000454

Example 4:

The average number of goals per match of Messi in the World Cup Soccer is approximately 2.5. Find the probability of 0, 1 or 2 goals of Messi per match in the World Cup.

Solution:

Because the average event rate is 2.5 goals per match, λ = 2.5.

$$P(k \text{ goals in a match}) = \frac{2.5^k e^{-2.5}}{k!}$$

$$P(k = 0 \text{ goals in a match}) = \frac{2.5^0 e^{-2.5}}{0!} = \frac{e^{-2.5}}{1} \approx 0.082$$

$$P(k = 1 \text{ goal in a match}) = \frac{2.5^1 e^{-2.5}}{1!} = \frac{2.5e^{-2.5}}{1} \approx 0.205$$

$$P(k = 2 \text{ goals in a match}) = \frac{2.5^2 e^{-2.5}}{2!} = \frac{6.25e^{-2.5}}{2} \approx 0.257$$

<u>Non-Stationary Poisson Process (NSPP):</u>

Assuming that, a Poisson process has a fixed and constant rate λ over all time limits to its applicability. In other words, λ doesn't change over time this is known as a time-stationary or time-homogenous Poisson process or just simply a stationary Poisson process.

But, questions is what happens when the arrival rate is non-stationary, i.e. the arrival rate λ(t) a function of time t ?

PR CHAND

It turns out that a stationary Poisson process with arrival rate 1 can be transformed into a non-stationary Poisson process with any time-dependent arrival rate.

The NSPP is useful for situations in which the arrival rate varies during the period of interest (time), for example, meal times for restaurants, phone calls during business hours, and orders for pizza delivery around 6 P. M. etc.

The key to working with an NSPP is the expected number of arrivals by time t, denoted by:

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

$E(N(t)) = \Lambda(t)$

$P(N(t)=k) = (e^{-\Lambda(t)} \Lambda(t)^k) / k!$

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 1:

- Suppose we are conducting a time study of a helicopter maintenance facility
- Our data indicates that the facility is busier in the morning than in the afternoon:
  - In the morning (0900 – 1300): expected interarrival time of 0.5 hours
  - In the afternoon (1300 – 1700): expected interarrival time of 2 hours

CHAND

a. What is the probability that 2 helicopters arrive between 1200 and 1400, given that 5 arrived between 0900 and 1200?
b. What is the expected number of helicopters to arrive between 1200 and 1400?

Solution:

Let’s say t = 0 correspond to 09:00

Therefore, the arrival rate λ(t) as a function of t (in hour) is:

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

Using this we can compute the integrated-rate function Λ(t), or the expected number of arrivals by time t:

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

$$\text{If} \quad t \in [0,4] : \quad \Lambda(t) = \int_0^t 2\, ds$$

$$\Lambda(t) = 2[s]_0^t = 2(t-0) = 2t$$

$$\text{If} \quad t \in [4,8] : \quad \Lambda(t) = \int_0^4 2\, ds + \int_4^t \frac{1}{2}\, ds$$

$$\Lambda(t) = 2[s]_0^4 + \frac{1}{2}[s]_4^t$$

$$= 2(4-0) + \frac{1}{2}(t-4)$$

$$= 2\times 4 + \frac{t}{2} - \frac{4}{2}$$

$$= 8 + \frac{t}{2} - 2$$

$$= \frac{t}{2} + 6$$

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

a) Given $Pr[N(3) - N(0) = 5]$

$Pr[N(5) - N(3) = 2]$

Now,

$$\Lambda(t) = N(5) - N(3)$$

$$= \Lambda(5) - \Lambda(3)$$

$$= \frac{t}{2} + 6 - 2t$$

$$= \frac{5}{2} + 6 - 2\times3$$

$$= \frac{17}{2} - 6$$

$$\Lambda(t) = \frac{5}{2} = 2.5$$

$$P(k=2) = \frac{e^{-\Lambda(t)}\,\Lambda(t)^{k}}{k!} = \frac{e^{-2.5}\,(5/2)^{2}}{2!}$$

$$= \frac{(2.718)^{-2.5}\,(2.5)^{2}}{2} = \frac{0.082 \times 6.25}{2}$$

$$= \frac{0.5125}{2} = 0.26$$

Hence, the probability of 2 helicopter arrive between 12:00 and 14:00 is 0.26

b) $E(N(5) - N(3))$

$$= \Lambda(5) - \Lambda(3)$$

$$= \frac{t}{2} + 6 - 2t = \frac{5}{2} + 6 - 2\times 3 = \frac{17}{2} - 6$$

$$= 5/2$$

$$= 2.5$$

Hence, the expected number of helicopter will arrive between 12:00 to 14:00 is 2.5

Batch/Bulk Arrival:

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: $t0 \leq t1 \leq t2 \leq \cdot \cdot \cdot$, 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: $0 = t0 = t1 = t2 < 1 = t3 = t4 = t5 = t6 < 3 = t7 = t8 < t9 \cdot \cdot \cdot$ means that a batch of size 3 occurred 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 Cj (the jth customer; arrival time tj) is a member within some batch. As is underlying the so called inspection paradox, we are more likely 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.

Gathering Statistics:

Most simulation programming system includes 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 those commonly needed statistics are:

1. ***Counts:*** Giving the number of entities of a particular type or the number of times some event occurred.
2. ***Summary measures:*** Such as extreme values, mean values, and standard deviations.
3. ***Utilization:*** Defined as the fraction (or percentage) of time some entity is engaged.
4. ***Occupancy:*** Defined as the fraction (or percentage) of a group of entities in use on the average.
5. ***Transit times:*** defined as the time taken for an entity to move from one part to the system to some other part.
6. Distributions of important variables such as queue length or waiting times.

PRATIK CHAND

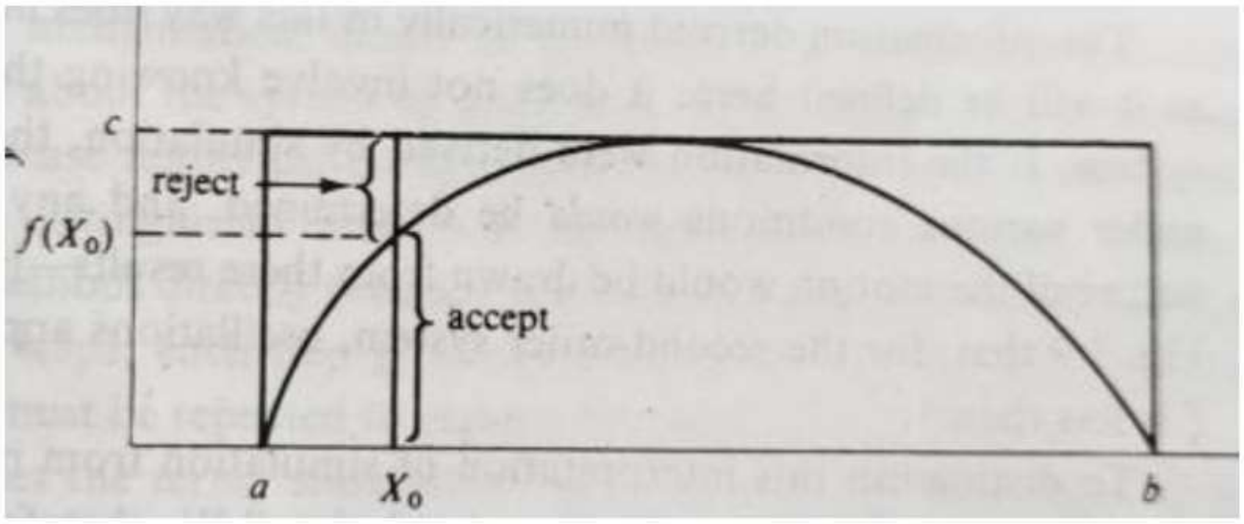
When there are stochastic effects operating in the system, all these system measures will fluctuate as a simulation proceeds, and the particular values reached at the end of the simulation are taken as estimates of the true values they are designed to measure.

Monte Carlo Method Simulation:

- Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models.
- Monte Carlo simulation can be used to tackle a range of problems in virtually every field such as finance, engineering, supply chain, and science.
- Monte Carlo simulation is also referred to as multiple probability simulation.
- Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. One of the basic examples of getting started with the Monte Carlo algorithm is the estimation of Pi.

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***.

PRATIK CHAND



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,

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

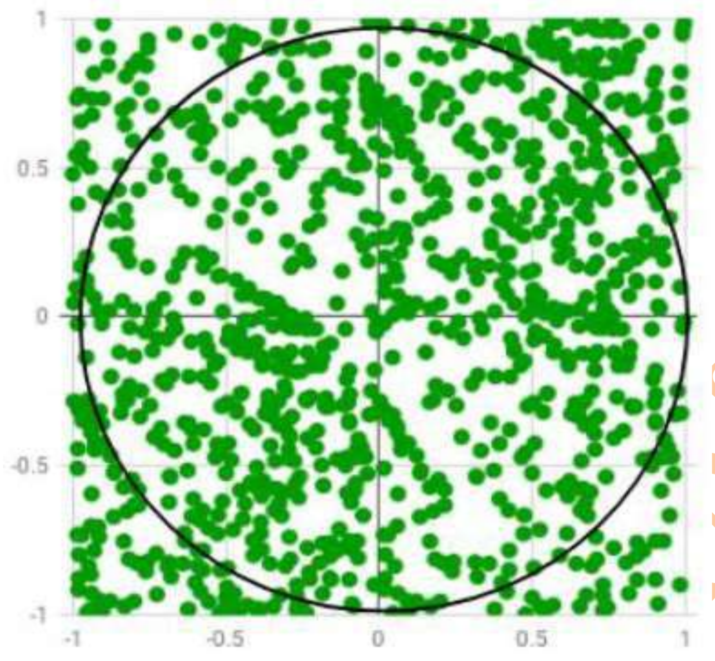
The accuracy improves as the number of ***N*** increases. When it is decided that the sufficient points has been taken, the value of integral is estimated by multiying ***n/N*** with the area of rectangle, ***c(b-a).***

Example: 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:

PRATIK CHAND



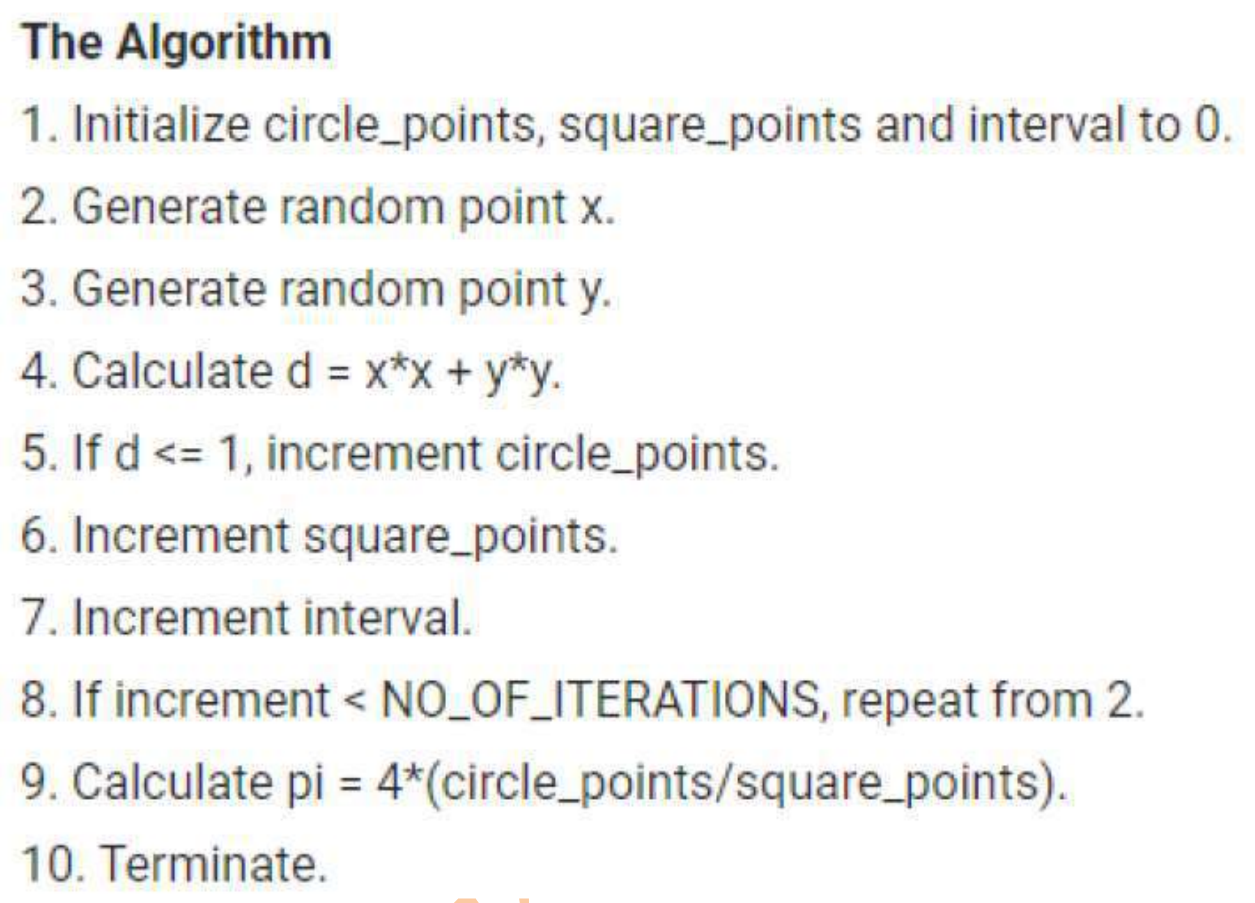
We know that area of the square is 1 unit sq while that of circle is $\pi * \left(\frac{1}{2}\right)^2 = \frac{\pi}{4}$. Now for a very large number of generated points,

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

that is,

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

PRATIK CHAND



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.

PRATIK

End of Unit-2