

CONTINUOUS SYSTEM MODELS

A continuous system is one in which the predominant activities of the system cause smooth changes in the attributes of the system entities. When such a system is modeled mathematically, the variables of the model representing the attributes are controlled by continuous functions. More generally in continuous system the relationships described the rates at which attributes change so that the model consists of differential equations.

Continuous simulation is something that can only really be accomplished with an analog computer. Using a digital computer one can approximate a continuous simulation by making the time step of the simulation sufficiently small so there are no transitions within the system between time steps. The premise for a continuous simulation is that there is a continuous time flow and the simulation is stepped in time increments.

Differential equations, both linear, nonlinear, ordinary and partial occur repeatedly in scientific and engineering studies. The reasons for this prominence is that most physical and chemical processes involve rates of change, which require differential equations for their mathematical description.

The simplest differential equation models have one or more linear differential equations with constant coefficients. It is then often possible to solve the model without the use of simulations. Even, so the labor involved may be so great that it is preferable to use simulation techniques. However, when nonlinearities are introduced into the model, if frequently becomes impossible or, at least, very difficult to solve the models. Simulation method of solving the models do not change fundamentally when nonlinearities occur. The methods of applying simulation to continuous models can therefore be developed by showing their application to models where the differential equations are linear and have constant coefficients, and then generalizing to more complex questions.

CONTINU

Continuous system by e continuously

Typically, w Unit I) that state variabl deferential e analytically

The simples linear differ then often simulation. i is preferable nonlinearitie becomes imp systems. Th the differen coefficients,

Examples o

Example: C population population t

the prey wi certain a > 0 due to inter proportional

$$x'(t) = \epsilon$$

Without pre
 $y' = -cy$ for a

Their birth finally find fo

CONTINUOUS SIMULATION

Continuous simulation concerns the modeling over time of a system by a representation in which state variables change continuously with respect to time.

Typically, we use differential equations (already discussed in Unit I) that give relationships for the rates of change of the state variables with time. So, how do we solve these systems of differential equations? In very easy cases these can be solved analytically otherwise solved numerically.

The simplest differential equation models have one or more linear differential equations with constant coefficients. It is then often possible to solve the model without the use of simulation. Even so, the labor involved may be so great that it is preferable to use simulation techniques. However, when nonlinearities are introduced into the model, it frequently becomes impossible or, at least, very difficult to model these systems. The methods of applying simulation to models where the differential equations are linear and have constant coefficients, and then generalizing to more complex equations.

Examples of Continuous Simulation

Example: Consider an easy predator-prey model. Let the prey population at time t be given by $x(t)$, and the predator population by $y(t)$. Assume that, in the absence of predators, the prey will grow exponentially according to $x' = ax$ for a certain $a > 0$. We also assume that the death rate of the prey due to interaction is proportional to $x(t)y(t)$, with a positive proportionality constant. So:

$$x'(t) = ax(t) - bx(t)y(t)$$

Without prey, predators will die exponentially according to $y' = -cy$ for a certain $c > 0$.

Their birth strongly depends on both population sizes, so we finally find for a certain $d > 0$:

$$y'(t) = -cy(t) + dx(t) y(t)$$

$$x'(t) = ax(t) - bx(t) y(t)$$

$$y'(t) = -cy(t) + dx(t) y(t)$$

We immediately see that both $(e^{at}, 0)$ and $(0, e^{-ct})$ are solution of $(x(t), y(t))$. From this system we find that for every solution we must have

$$x' \frac{c}{x} - a + y' \frac{a}{y} - b = 0$$

$$x' \frac{c}{x} - a + y' \frac{a}{y} - b = 0$$

Integrating both sides of above equation gives us $c \log x(t) - dx(t) + a \log y(t) - by(t) = \text{constant}$. Solutions for $a = b = c = d = 1$.

The given model is considered very simple. Why? the integrating we did two above is possible, rest of the world has no influence, no randomness involved. Usually we cannot find closed-form solutions for the system of differential equations. How do we deal with this problem? Solve numerically.

Integrating both sides of above equation gives us

$$c \log x(t) - dx(t) + a \log y(t) - by(t) = \text{constant}$$

$$\text{Solutions for } a = b = c = d = 1$$

ANALOG COMPUTERS

Historically, continuous system simulation was in general use for studying complex systems long before discrete system simulation was similarly applied. The main reason was that, before the general availability of digital computers, needed for the numerical computation of discrete system simulation, there existed devices whose behavior is equivalent to a mathematical operation such as addition or integration. Putting together combinations of such devices in a manner specified by a mathematical model of a system

allowed the system devices gave control readily to the simulation devices have been a general a technique, *analog computers*, or differential equation models based on a specific example between electrical and an analog computer being based on a physical model.

The most widely used analog computer, based on current amplifiers, the computer are equipped with operational amplifier appropriate circuits, input voltages, each produce a voltage response. Different scale factor coefficients of the called summers. An integrator for which time of a single input voltages. All voltages to the sign of the variable of the model, it is so which is an amplifier the sign of the input.

Electronic analog computers for reasons. It is difficult to obtain a voltage beyond a few assumptions are made operational amplifiers do not solve

allowed the system to be simulated. By their nature, the devices gave continuous outputs and so lent themselves readily to the simulation of continuous systems.¹ Specific devices have been created for particular systems but with so general a technique, it has been customary to refer to them as *analog computers*, or, when they are primarily used to solve differential equation models, as *differential analyzers*. Physical models based on analogies were described in Chapter 1. The specific example discussed there described the analogy between electrical and mechanical systems. Simulation with an analog computer, however, is more properly described as being based on a mathematical model than as being a physical model.

The most widely used form of analog computer is the electronic analog computer, based on the use of high gain dc (direct current) amplifiers, called operational amplifiers. Voltages in the computer are equated to mathematical variables, and the operational amplifiers can add and integrate the voltages. With appropriate circuits, an amplifier can be made to add several input voltages, each representing a variable of the model, to produce a voltage representing the sum of the input variables. Different scale factors can be used on the inputs to represent coefficients of the model equations. Such amplifiers are called summers. Another circuit arrangement produces an integrator for which the output is the integral with respect to time of a single input voltage or the sum of several input voltages. All voltages can be positive or negative to correspond to the sign of the variable represented. To satisfy the equations of the model, it is sometimes necessary to use a sign inverter, which is an amplifier designed to cause the output to reverse the sign of the input.

Electronic analog computers are limited in accuracy for several reasons. It is difficult to carry the accuracy of measuring a voltage beyond a certain point. Secondly, a number of assumptions are made in deriving the relationships for operational amplifiers, none of which is strictly true; so, amplifiers do not solve the mathematical model with complete

accuracy. A particularly troublesome assumption is that there should be zero output for zero input. Another type of difficulty is presented by the fact that the operational amplifiers have a limited dynamic range of output, so that scale factors must be introduced to keep within the range. As a consequence, it is difficult to maintain an accuracy better than 0.1 % in an electronic analog computer. Other forms of analog computers have similar problems and their accuracies are not significantly better.

A digital computer is not subject to the same type of inaccuracies. Virtually any degree of accuracy can be programmed and, with the use of floating-point representation of numbers, an extremely wide range of variations can be tolerated. Integration of variables is not a natural capability of a digital computer, as it is in an analog computer, so that integration must be carried out by numerical approximations. However, methods have been developed which can maintain a very high degree of accuracy.

A digital computer also has the advantage of being easily used for many different problems. An analog computer must usually be dedicated to one application at a time, although time-sharing sections of an analog computer has become possible.

In spite of the widespread availability of digital computers, many users prefer to use analog computers. There are several considerations involved. The analog representation of a system is often more natural in the sense that it directly reflects the structure of the system; thus simplifying both the setting-up of a simulation and the interpretation of the results. Under certain circumstances, an analog computer is faster than a digital computer, principally because it can be solving many equations in a truly simultaneous manner ; whereas a digital computer can be working only on one equation at a time, giving the appearance of simultaneity by interlacing the equations. On the other hand, the possible disadvantages of analog computers, such as limited accuracy and the need to dedicate the computer to one problem, may not be significant.

ANALOG METHODS

The general method by which analog computers are applied can be demonstrated using the second-order differential equation that has already been discussed:

$$M\ddot{x} + D\dot{x} + Kx = KF(t)$$

Solving the equation for the highest order gives

$$M\ddot{x} = KF(t) - D\dot{x} - Kx$$

Suppose a variable representing the input $F(t)$ is supplied, and assume for the time being that there exist variables representing $-x$ and $-\dot{x}$. These three variables can be scaled and added with a summer to produce a voltage representing $M\ddot{x}$. Integrating this variable with a scale factor of $\frac{1}{M}$ produces

\ddot{x} . Changing the sign produces $-\dot{x}$, which supplies one of the variables initially assumed; and a further integration produces $-x$, which was the other assumed variable. For convenience, a further sign inverter is included to produce $+x$ as an output.

A block diagram to solve the problem in this manner is shown in Fig. 2.1. The symbols used in the figure are standard symbols for drawing block diagrams representing along computer arrangements. The circles indicate scale factors applied to the variables. The triangular symbol at the left of the figure represents the operation of adding variables. The triangular symbol with a vertical bar represents an integration, and the one containing a minus sign is a sign changer.

$$M\ddot{x} + D\dot{x} + Kx = KF(t)$$

The addition on the left, with its associated scaling factors, corresponds to the addition of the variables representing the three forces on the wheel, producing a variable representing $M\ddot{x}$. The scale is changed to produce \dot{x} , and the result is integrated twice to produce both x and $-x$. Sign changers are introduced so that variables of the correct sign can be fed back to the adder, and the output can be given in convenient form.

With an electronic analog computer, the variables that have been described would be voltages, and the symbols would represent operational amplifiers arranged as adders, integrators, and sign changers. Figure 2.1 would then represent how the amplifiers are interconnected to solve the equation. It should be pointed out, however, that there can be several ways of drawing a diagram for a particular problem, depending upon which variables are of interest, and on the size of the scale factors.

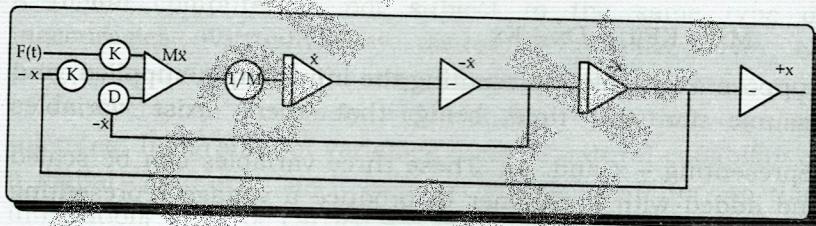


Fig. 2.1: Diagram for the automobile suspension problem

When a model has more than one independent variable, a separate block diagrams drawn for each independent variable and, where necessary, inter connections are made between the diagrams. As an example, Fig. 2.2 shows a block diagram for solving the model of the liver. There are three integrators, shown at the bottom of the figure. Reading from left to right, they solve the equations for x_1 , x_2 , and x_3 . Interconnections between the three integrators, with sign changers where necessary, provide inputs that define the differential coefficients of the three variables. The first integrator, for example, is solving the equation.

$$\dot{x}_1 = -k_{12}x_1 + k_{21}x_2$$

The second integrator is solving the equation

$$\dot{x}_2 = k_{12}x_1 - (k_{21} + k_{23})x_2$$

In this case, the variable x_2 is being used twice as an input to the integrator, so that the two coefficient k_{21} and k_{23} can be changed independently. The last integrator solves the equation

$$\dot{x}_3 = k_{23}x_2$$

HYBR

The scope by dev computer multipli were ex improvising amplifie easier to Among t compute Boolean and oper

The term of tradit continu nonlinear perform connotat usually construc compute standarc

Hybrid c mainly c

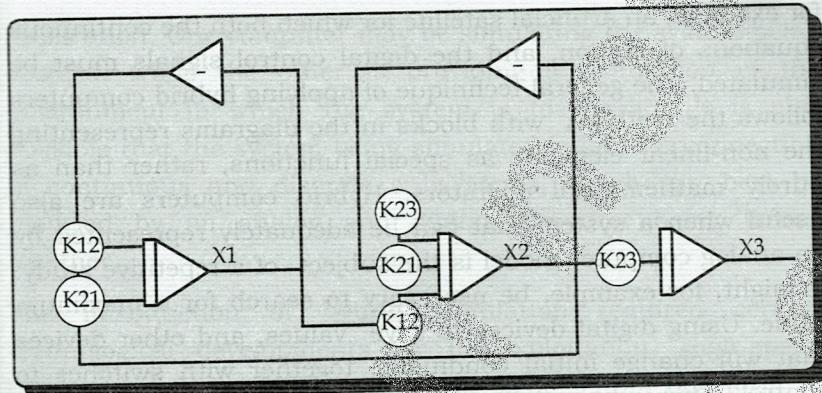


Fig: 2.2: Analog computer model of the liver

HYBRID COMPUTERS

The scope of analog computers has been considerably extended by developments in solid-logic electronic devices. Analog computers always used a few nonlinear 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. The general technique of applying hybrid computers follows the methods with blocks in the diagrams representing the non-linear elements as special functions, rather than as purely mathematical operators. 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. It might, for example, be necessary to search for a maximum value. Using digital devices to save values, and other devices that will change initial conditions, together with switches to control runs, a hybrid computer can be arranged to carry out large portions of the study without human intervention.

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 languages contain macro-instructions that carry out the action of adders, integrators, and sign changers. A program is written to link together these macro-instructions, in essentially the same manner as operational amplifiers are connected in analog computers.

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 extensive use.

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 use an analog or digital computer for system simulation. However, there are times when an analog and digital computer are combined to provide a hybrid simulation. The form taken by hybrid simulation depends upon the

applied
studied
environ
possibl
of cont
modele
togethe
The ir
technol
convert
represe
availabi
easier,
dedicate
generall
analog
purpose
of digit
analog c

FEED

A signifi
a coupli
The ter
home ho
example
purpose
measure
temperat
furnace
feedback
only two
An exam
control is

application. One computer may be simulating the system being studied, while the 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 best 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. As a practical matter, 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 functionally distinct analog and digital computers are linked together for the purpose of simulation. It should not be confused with the use of digital elements added to the operational amplifiers of an analog computer when describing hybrid computers.

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 states, either the furnace is on or off.

An example of a feedback system in which there is continuous control is the aircraft system and illustrated in Fig 2.3.

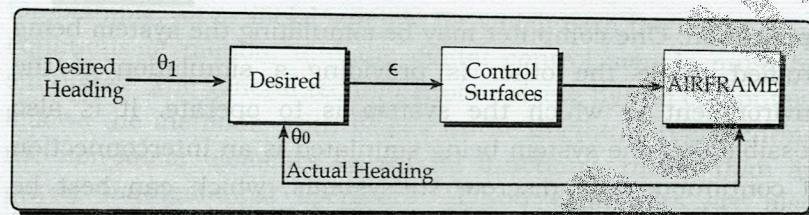


Fig. 2.3: An Aircraft under autopilot control

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 surfaces, since change of heading will then affect the signal being used to control the heading. The difference between the desired heading θ_1 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 by ϵ .

Suppose the control surface angle is made directly proportional to the error signal. The force changing the heading is then proportional to the error and consequently, it diminishes as the aircraft is suddenly asked to change to a new heading θ_i . The subsequent changes are illustrated in figure 2.4. The upper curves represent the control surface angle and the lower curves represent the aircraft heading.

Consider first the solid curves of fig 2.4. Upon receipt of a signal to change direction, the error signal, and therefore the control surface angle, suddenly takes a non-zero value. The aircraft heading, shown in the lower curves of fig 2.4, inertia, it takes time to respond. As the aircraft turns, the control surface angle decreases so that less force is applied as the aircraft approaches the required heading. Ultimately, when the aircraft reaches the required heading, the control surface angle will be zero but the inertia of the aircraft can carry it beyond the desired heading. As a result, the control surface turns in the opposite direction in order to bring the aircraft back from its overshoot. The correction from the overshoot produces an undershoot and the motion follows a series of oscillations of decreasing amplitude as illustrated in Fig. 2.4.

Suppose the system signal there is a a of Fig. 2.4 represents restoring force is 1 but, corresponding aircraft will oscillate in Fig 4-6. The feed greater amplification correction force. U amplification to be the initial error; i correction become unstable because o

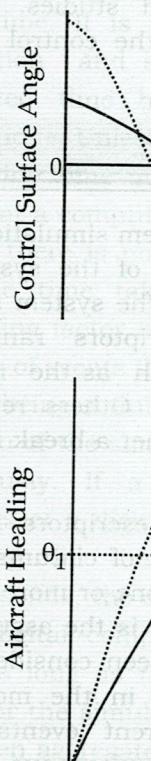


Fig 2.4:

Suppose the system is changed so that for the same size error signal there is a larger control surface angle. The dotted curves of Fig. 2.4 represent this case. for the same conditions, the restoring force is larger and the aircraft respond more rapidly but, correspondingly, the initial overshoot will be larger. The aircraft will oscillate more widely and readily, as is illustrated in Fig 4-6. The feedback loop in the second case is said to have greater amplification, since the same error produces a larger correction force. Under certain conditions, it is possible for the amplification to be so greater that the initial overshoot exceeds the initial error; in which case, the undershoot from the correction becomes even larger and the system becomes unstable because of ever-increasing oscillations.

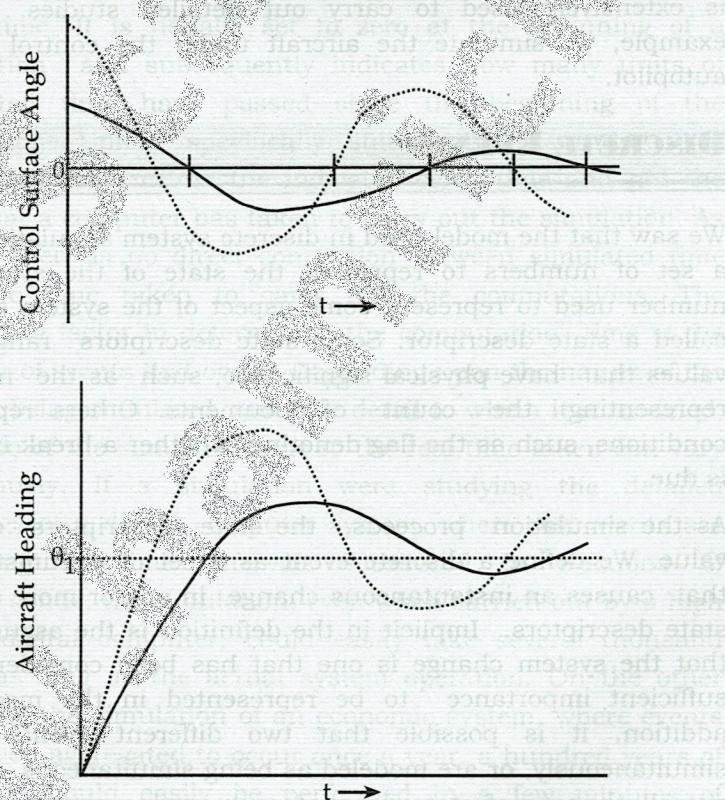


Fig 2.4: Aircraft response to autopilot system

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

Generally, negative feedback is a stabilizing influence, although, as we have just seen, it can become strong enough to be a destabilizing influence. There are occasions when a system experiences positive feedback, in which case the force tends to increase the deviation. Any prolonged positive feedback will make a system unstable.

Many simulation studies of continuous systems are concerned primarily with the study of servomechanisms, which is the general name given to devices that rely upon feedback for their operation. The field of control theory provides the theoretical background with which to design such systems, but simulation is extensively used to carry out detailed studies. As an example, we simulate the aircraft under the control of the autopilot.

DISCRETE EVENTS

We saw that the model used in discrete system simulation has a set of numbers to represent the state of the system. A number used to represent some aspect of the system state is called a state descriptor. Some state descriptors range over values that have physical significance, such as the number representing the count of documents. Others represent conditions, such as the flag denoting whether a break in work is due.

As the simulation proceeds, the state descriptors change value. We define a discrete event as a set of circumstances, that causes an instantaneous change in one or more system state descriptors. Implicit in the definition is the assumption that the system change is one that has been considered of sufficient importance to be represented in the model. In addition, it is possible that two different events occur simultaneously, or are modeled as being simultaneous, so that not all changes of state descriptors occurring simultaneously necessarily belong to a single event.

The term simultaneous, of course, refers to the occurrence of the changes in the system, and not to when the changes are made in the model- in which, of necessity, the changes must be made sequentially. A system simulation must contain a number representing time. The simulation proceeds by executing all the changes to the system descriptors associated with each event, as the events occur in chronological order. The way in which events are selected for execution, particularly when there are simultaneous events, is an important aspect of programming simulations.

REPRESENTATION OF TIME

The passage of time is recorded by a number referred to as clock time. It is usually set to zero at the beginning of a simulation and subsequently indicates how many units of simulated time have passed since the beginning of the simulation. Unless specifically stated otherwise, the term simulation time means the indicated clock time and not the time that a computer has taken to carry out the simulation. As a rule, there is no direct connection between simulated time and the time taken to carry out the computations. The controlling factor in determining the computation time is the number of events that occur. Depending upon the nature of the system being simulated, and the detail to which it is modeled, the ratio of the simulated time to the real time taken can vary enormously. If a simulation were studying the detailed workings of a digital computer system, where real events are occurring in time intervals measured in fractions of microseconds, the simulation, even when carried out by a high speed digital computer, could easily take several thousand times as long as the actual system operation. On the other hand, for the simulation of an economic system, where events have been aggregated to occur once a year, a hundred years of operation could easily be performed in a few minutes of calculations.

Two basic methods exist for updating clock time. One method is to advance the clock to the time at which the next event is due to occur. The other method is to advance the clock by small (usually uniform) intervals of time and determine at each interval whether an event is due to occur at that time. The first method is referred to as event-oriented, and the second method is said to be interval-oriented. Discrete system simulation is usually carried out by using the event-oriented method, while continuous system simulation normally uses the interval-oriented method.

It should be pointed out, however, that no firm rule can be made about the way time is represented in simulations for discrete and continuous systems. An interval-oriented program will detect discrete changes and can therefore simulate discrete systems, and an event-oriented program can be made to follow continuous changes by artificially introducing events that occur at regular time intervals. Further, the event-oriented method is not necessarily faster than the interval-oriented method for discrete systems..

Another approach to representing the passage of time has been called significant event simulation. The method is applicable to continuous systems in which there are quiescent periods. The interval between events in the event-oriented approach is, of course, a quiescent period, but it involves having the model's representation of the system activities create a notice of the event that terminates the interval. The significant event approach assumes that simple analytic functions, such as polynomials of low order, can be used to project the span of a quiescent period. The event that ends the period could be any one of several alternatives, each of which has a projected span. The significant event is the one with the least span. Determining this event, by simple comparisons of the projections, allows the clock to be updated by an extended period of time; achieving the same thing would otherwise have cost the effort of executing the updating of many fruitless intervals of fixed size.

MODEL OF ARRIVAL PROCESS

POISSON PROCESS

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 $0, 1, 2, \dots$. The counting processes $\{N(t), t \geq 0\}$, is said to be a Poisson process with mean rate λ if the following assumptions are fulfilled:

1. Arrivals occur one at a time.
2. $\{N(t) \geq 0\}$ 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 period.
3. $\{N(t), t \geq 0\}$ has independent increments: The number of arrivals during non-overlapping 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.

If arrivals occur according to a Poisson process, meeting the three preceding assumptions, it can be shown that the probability that $N(t)$ is equal to n is given by

$$P[N(t) = n] = \frac{e^{-\lambda t} (\lambda t)^n}{n!} \quad \text{for } t \geq 0 \text{ and } n = 0, 1, 2, \dots \quad N(t)$$

has the Poisson distribution with parameter $\alpha = \lambda t$.

Thus, its mean and variance are given by

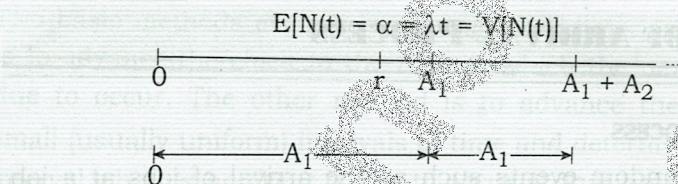


Fig 2.5: Arrival Process

For any time s and t such that $s < t$, assumption of stationary increments implies that the random variable $N(t) - N(s)$, representing the number of arrivals in the interval from s to t , is also poisson-distributed with mean $\lambda(t-s)$. Thus,

$$P[N(t) - N(s) = n] = \frac{e^{-\lambda(t-s)} [\lambda(t-s)]^n}{n!} \quad \text{for } n = 0, 1, 2, \dots$$

$$\text{and, } E[N(t) - N(s)] = \lambda(t-s) = V[N(t) - N(s)]$$

Now, consider the time at which arrivals occur in a poisson process. Let the first arrival occur at time A_1 , the second occur at time $A_1 + A_2$, and so on, as shown in Figure 2.5. Thus, A_1, A_2, \dots are successive interarrival times. The first arrival occurs after time t if and only if there are no arrivals in the interval $[0, t]$, so it is seen that

$$[A_1 > t] = \{N(t) = 0\}$$

and, therefore,

$$P(A_1 > t) = P[N(t) = 0] = e^{-\lambda t}$$

the last equality. Thus, the probability that the first arrival will occur in $[0, t]$ is given by

$$P(A_1 \leq t) = 1 - e^{-\lambda t}$$

which is cdf for an exponential distribution with parameter λ . Hence, A_1 is distributed exponentially with mean $E(A_1) = 1/\lambda$. It also be shown that all inter-arrival time, A_1, A_2, \dots are exponentially distributed and independent with mean $1/\lambda$. As an Alternatively definition of a Poisson process, it can be shown that, if inter-arrival time are distributed exponentially and independently, then the number of arrivals by time t say $N(t)$, meets the three previously mentioned assumptions and, therefore, is a Poisson process.

Recall that the exponential distribution is memory less—that is, the probability of a future arrival in a time interval of length s is independent of the time of the last arrival. The probability of the arrival depends only on the length of the time interval, s , thus, the memory less property is related to the properties of independent and stationary increments of the Poisson process.

Example

The jobs at a machine shop arrive according to a Poisson process with a mean of $\lambda = 2$ jobs per hour, therefore, the interarrival times are distributed exponentially, with the expected time between arrivals being $E(A) = 1/\lambda = \frac{1}{2}$ hour.

PROPERTIES OF A POISSON PROCESS

Several properties of the Poisson process, discussed by Ross [2002] and others, are useful in discrete-system simulation. The first of these properties concerns random splitting, consider a Poisson process $\{N(t), t \geq 0\}$ having rate λ , as represented by the Figure 2.5



Fig 2.5: Random Splitting

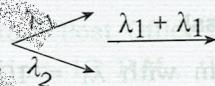


Fig 2.6: Pooled Splitting

Suppose that, each time an event occurs, it is classified as either a type I or a type II event. Suppose further that each event is classified as a type I event with probability p and type II event with probability $1 - p$, independently of all other events.

Let $N_1(t)$ and $N_2(t)$ be random variables that denote, respectively, the number of type I and type II events occurring in $[0, t]$. Note that $N(t) = N_1(t) + N_2(t)$. It can be shown that $N_1(t)$ and

$N_1(t)$ and $N_2(t)$ are both Poisson processes having rates λ_p and $\lambda(1 - p)$, as shown in Figure 2.6. Furthermore, it can be shown that the two processes are independent.

Example (Random Splitting)

Suppose that job arrive at a shop in accordance with a Poisson process having rate λ . Suppose further that each arrival is marked "High priority" with probability $1/3$ and "low priority" with probability $2/3$. Then a type I event would correspond to a high-priority arrival and a type II event would correspond to a low-priority arrival. If $N_1(t)$ and $N_2(t)$ are as just defined, both variables follow the Poisson process, with rates $\lambda/3$ and $2\lambda/3$, respectively.

Example

The rate is $\lambda = 3$ per hour. The probability that no high-priority jobs will arrive in a 2 hours period is given by the Poisson distribution with parameter $a = \lambda pt = 2$. Thus,

$$P(0) = \frac{e^{-2}}{0!} = 0.135$$

Now, consider the opposite situation from random splitting, namely the pooling of two arrival streams. The process of interest is illustrated in Figure 5.26. It can be shown that, if $N_i(t)$ are random variables representing independent Poisson processes with rates λ_i , for $i = 1$ and 2 , then $N(t) = N_1(t) + N_2(t)$ is a Poisson process with rate $\lambda_1 + \lambda_2$.

Example (Pooled Process)

A Poisson arrival stream with $\lambda_1 = 10$ arrivals per hour is combined (or pooled) with a Poisson arrival stream with $\lambda_2 = 17$ arrivals per hour. The combined process is a Poisson Process with $\lambda = 27$ arrivals per hour.

Non-stationary Poisson Process

If we keep the Poisson Assumptions 1 and 3, but drop Assumption 2 (Stationary increments) then we have a non stationary Poisson process (NSPP), which i.e. 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

$(1-p)$, at the interest, including meal time for restaurants, phone calls during business hours, and orders for pizza delivery around 6 P.M.

The key to working a 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 Λ we have $\Lambda(t) = \lambda t$, as expected.

Let T_1, T_2, \dots be the arrival times of stationary Poisson process $N(t)$ with $\lambda = 1$, and let T_1, T_2, \dots be the arrival times for a NSPP $N(t)$ with arrival rate $\lambda(t)$. The fundamental relationship for working with NSPPs is the following:

$$T_i = \Lambda(T_i)$$

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

In 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. Unit 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 a NSPP $N(t)$ with rate function

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

The expected number of arrivals by time t is therefore,

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

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

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

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

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

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

BATCH 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: $t_0 \leq t_1 \leq t_2 \leq \dots$, where $\lim_{n \rightarrow \infty} t_n = \infty$. 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 $0 = t_0 = t_1 = t_2 < 1 = t_3 = t_4 = t_5 = t_6 < 3 = t_7 = t_8 < t_9 \dots$ 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 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 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. Of intrinsic interest is the position J of this randomly chosen customer within the batch. In the above example, the second batch (of size 4) has four positions within it: first, second, third and fourth, as indexed by the four integers (3, 4, 5, 6) corresponding to the four arrival times $t_3 = t_4 = t_5 = t_6$. It helps to imagine the batch as a bus with one long line of seats indexed 1, ..., v where v is the batch size. Customers occupy the seats in the order of the index j of arrival epochs t_j . In our $t_3 = t_4 = t_5 = t_6$ example, $v = 4$ and C_3 sits in seat 1, C_4 sits in seat 2, C_5 sits in seat 3, and C_6 sits in seat 4. We shall assume for now that batch sizes $\{v_n\}$ are i.i.d. with distribution

$P(v = k) = p_k$, $k \geq 1$, having finite mean $E(v) = \frac{1}{\alpha}$. Then the distribution of J (the position within a batch of a randomly chosen customer) is given by

$$\begin{aligned} P(J = k) &= \frac{P(v \geq k)}{E(v)} \\ &= \alpha \sum_{j=k}^{\infty} p_j, k \geq 1 \end{aligned}$$

(1) is to be interpreted as the long run proportion of customers who are k th within a batch; thus, from the renewal reward theorem, can be computed as the expected number of customers within a batch ("cycle") who are in position k , divided by the expected batch size: For fixed position k , there can be at most one customer within a batch who is in position k , and this will happen iff the batch is at least of size k . So, the reward during a batch of size v is given by $R = I\{v \geq k\}$, yielding $E(R) = \alpha P(v \geq k)$.

This is analogous to the continuous-time equilibrium distribution of v (density given by $f_e(x) = \alpha P(v > x)$, $x \geq 0$), but now we are using discrete time to average over, and J always has a discrete distribution (see Remark 3 below). For example, in the deterministic case with $P(v = 2) = 1$, J has the discrete

uniform distribution over the integers {1, 2}, whereas f_e is the uniform density over the interval (0, 2). To compute the expected value, $E(J)$, we interpret it as the long run average (over all customers) position within a batch. From the renewal reward theorem we can compute this as the expected value of the sum of all the positions over a batch divided by the expected batch size: Note that over a batch of size v , the sum of the positions 1 of all v customers is given by

$$R = 1 + \dots + v = \frac{v(v+1)}{2}$$

Thus,

$$E(J) = \frac{E(v(v+1))}{2E(v)} \quad \dots (2)$$

which can also be computed directly:

$$E(J) = \sum_{k=1}^{\infty} k P(J=k)$$

$$= \sum_{j=1}^{\infty} j \left(\alpha \sum_{j=k}^{\infty} p_j \right)$$

$$= \alpha \sum_{j=1}^{\infty} p_j \left(\sum_{k=1}^j k \right)$$

$$= \alpha \sum_{j=1}^{\infty} p_j ((j+1)/2)$$

$$= \frac{E(v(v+1))}{2E(v)}$$

Similarly, the total size of the batch containing a randomly chosen customer has distribution given by

$$\frac{kP(v=k)}{E(v)} = \alpha kp_k, k \geq 1, \quad \dots (3)$$

interpreted as the long run proportion of customers who are in a batch of size k . Once again, this follows from renewal reward, with $R = kI\{v=k\}$: Batches of size k each contain exactly k customers from a batch of size k (no other batches are

- (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.

PROBABILITY AND MONTE CARLO SIMULATION

Simulation can also be defined as a technique of performing sampling experiments on the model of the system. This is called stochastic simulation and is a part of simulation techniques. Because sampling from a particular probability distribution involves the use of random numbers, stochastic simulation is sometimes called Monte Carlo Simulation. Historically, Monte Carlo method is considered to be a technique, using random or pseudo random numbers.

Monte-Carlo methods: It have been invented in the context of the development of the atomic bomb in the 1940's. They are a class of computational algorithms and can be applied to vast ranges of problems. They are not a statistical tool and rely on repeated random sampling. They provide generally approximate solutions and are used in cases where analytical or numerical solutions don't exist or are too difficult to implement.

Monte-Carlo methods generally follow the following steps:

1. Determine the statistical properties of possible inputs
2. Generate many sets of possible inputs which follows the above properties
3. Perform a deterministic calculation with these sets
4. Analyze statistically the results

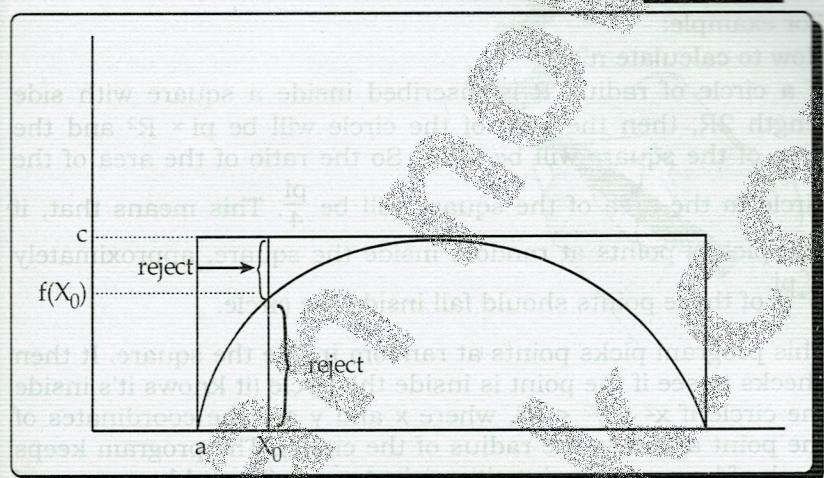


Fig: 2.7: The Monte Carlo Method

APPLICATIONS OF MONTE CARLO METHOD:

1. Numerical Integration
2. Error estimation
3. Optimization problems
4. Numerical Simulations

For example:

What is the probability to obtain either 3, 6 or 9 heads if one draws a coin ten times?

Binomial probability: $P = B(3; 10, 0.5) + B(6; 10, 0.5) + B(9; 10, 0.5) \approx 0.33$

Monte-Carlo simulation:

1. Given a random variable $y \sim U(0, 1)$, define "head" if $y < 0.5$, "tail" otherwise
2. Draw 10 random variables $x_i \sim U(0, 1)$, $i = 1, \dots, 10$
3. Count the number of heads H , and increment T if $H = 3, 6$, or 9
4. Repeat 2.-3. N times, with N reasonably large
5. The probability is approximately T/N

Note that this is an integration on a probability distribution, even if it is discrete!

For example:

How to calculate π ?

If a circle of radius R is inscribed inside a square with side length $2R$, then the area of the circle will be $\pi \times R^2$ and the area of the square will be $(2R)^2$. So the ratio of the area of the circle to the area of the square will be $\frac{\pi}{4}$. This means that, if you pick N points at random inside the square, approximately $N \cdot \frac{\pi}{4}$ of those points should fall inside the circle.

This program picks points at random inside the square. It then checks to see if the point is inside the circle (it knows it's inside the circle if $x^2 + y^2 < R^2$, where x and y are the coordinates of the point and R is the radius of the circle). The program keeps track of how many points it's picked so far (N) and how many of those points fell inside the circle (M).

π is then approximated as follows:

$$\pi = \frac{4M}{N}$$

Although the Monte Carlo Method is often useful for solving problems in physics and mathematics which cannot be solved by analytical means, it is a rather slow method of calculating π .



DISCUSSION EXERCISE

1. What is Poisson process?
2. Write short notes on discrete distributions.
3. What is differential equation? How can you model differential equation by analog method? Show with example.
4. What do you mean by hybrid simulation?
5. Differentiate between fixed interval and next event time representation.
6. Find the value of π using Monte Carlo method.
7. Define simulation clock.
8. Describe the symbols used in analog method and draw model of functioning of human liver using analog method.

