SIMULATION BASICS

Types of Simulation

- Static atau dynamics
- Stochastic atau deterministic
- Discrete event atau continuous

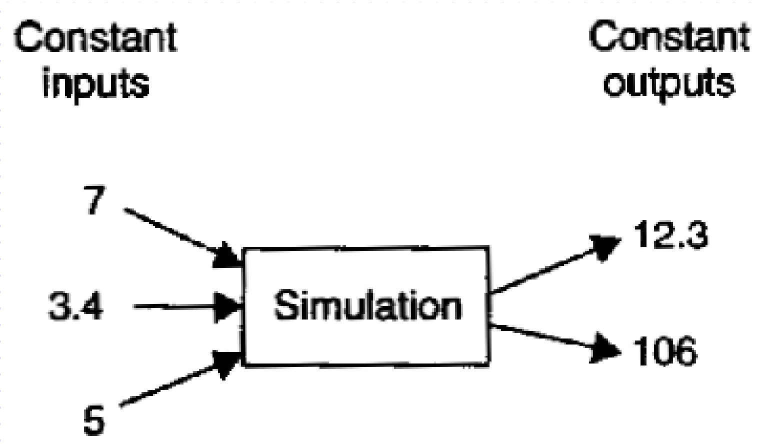
Static vs. Dynamic Simulation

- Simulasi statis adalah salah satu yang tidak didasarkan atas waktu
- Sering melibatkan sampel acak untuk menghasilkan hasil statistik, sehingga kadang-kadang disebut Monte Carlo Simulasi
- Simulasi Dinamis meliputi waktu yang terlalui
- Ini terlihat pada perubahan keadaan yang terjadi dari waktu ke waktu.
- Sebuah mekanisme jam bergerak maju dalam waktu dan variabel state diperbarui sebagai waktu advances.
- Simulasi dinamis cocok untuk menganalisis sistem manufaktur dan jasa

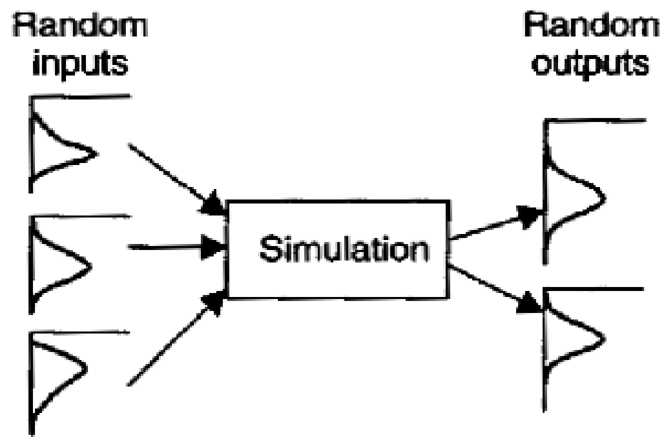
Stochastic vs. Deterministic Simulation

- Simulasi di mana satu atau lebih variabel input bersifat acak disebut sebagai simulasi stokastik atau probabilistik
- Sebuah simulasi stokastik menghasilkan output yang acak.
- Simulasi tidak memiliki komponen input yang acak yang dikatakan menjadi deterministik.
- Sebuah simulasi deterministik selalu akan menghasilkan hasil yang sama persis tidak peduli berapa kali dijalankan

Stochastic vs. Deterministic Simulation



(a) Deterministic simulation



(b) Stochastic simulation

Random Behavior

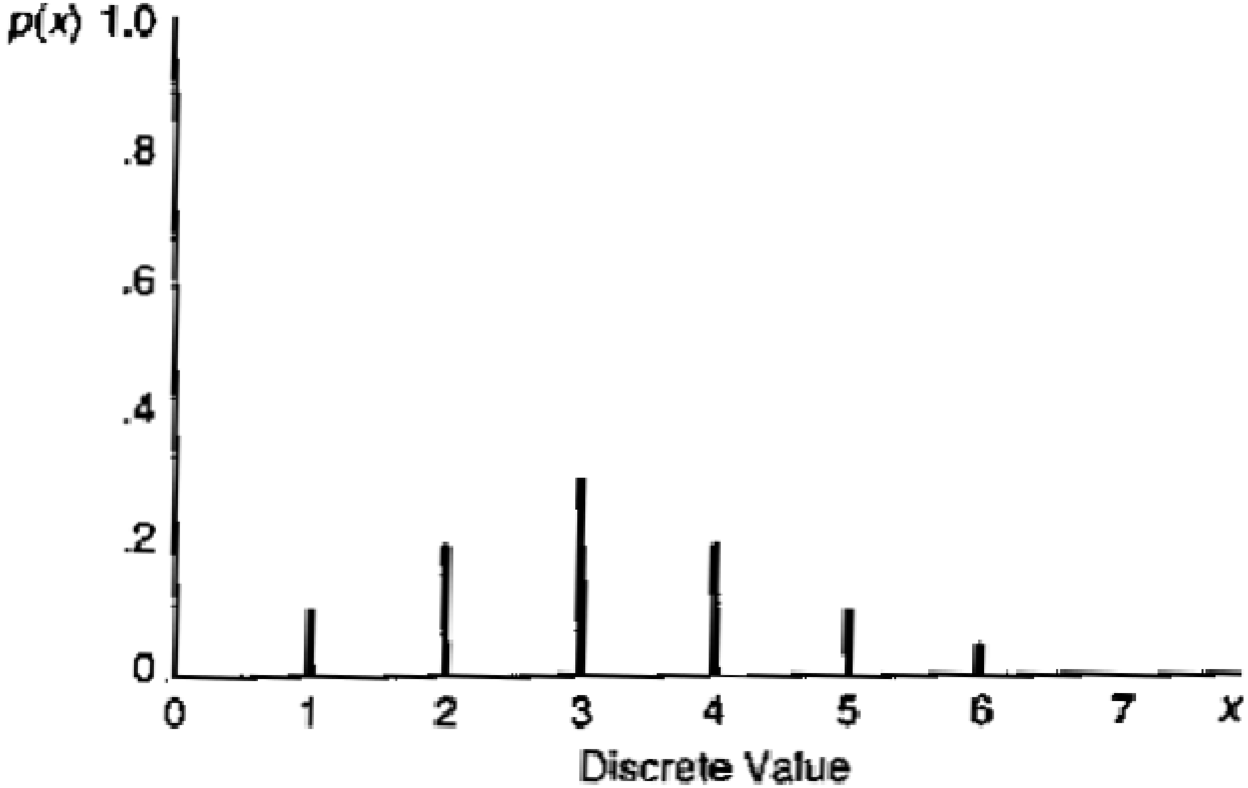
- Stochastic systems frequently have time or quantity values that vary within a given range and according to specified density, as defined by a probability distribution.
- Probability distributions are useful for predicting the next time, distance, quantity, and so forth when these values are random variables.
- Probability distributions are defined by specifying the type of distribution (normal, exponential or another type) and the parameters that describe the shape or density and range of the distribution.

Random Behavior

- A random variate is a value generated from a distribution.
- Probability distributions may be either discrete or continuous.
- A discrete distribution represents a finite or countable number of possible values.
  - The number of items in a lot
- A continuous distribution represents a continuum of values
  - Processing time

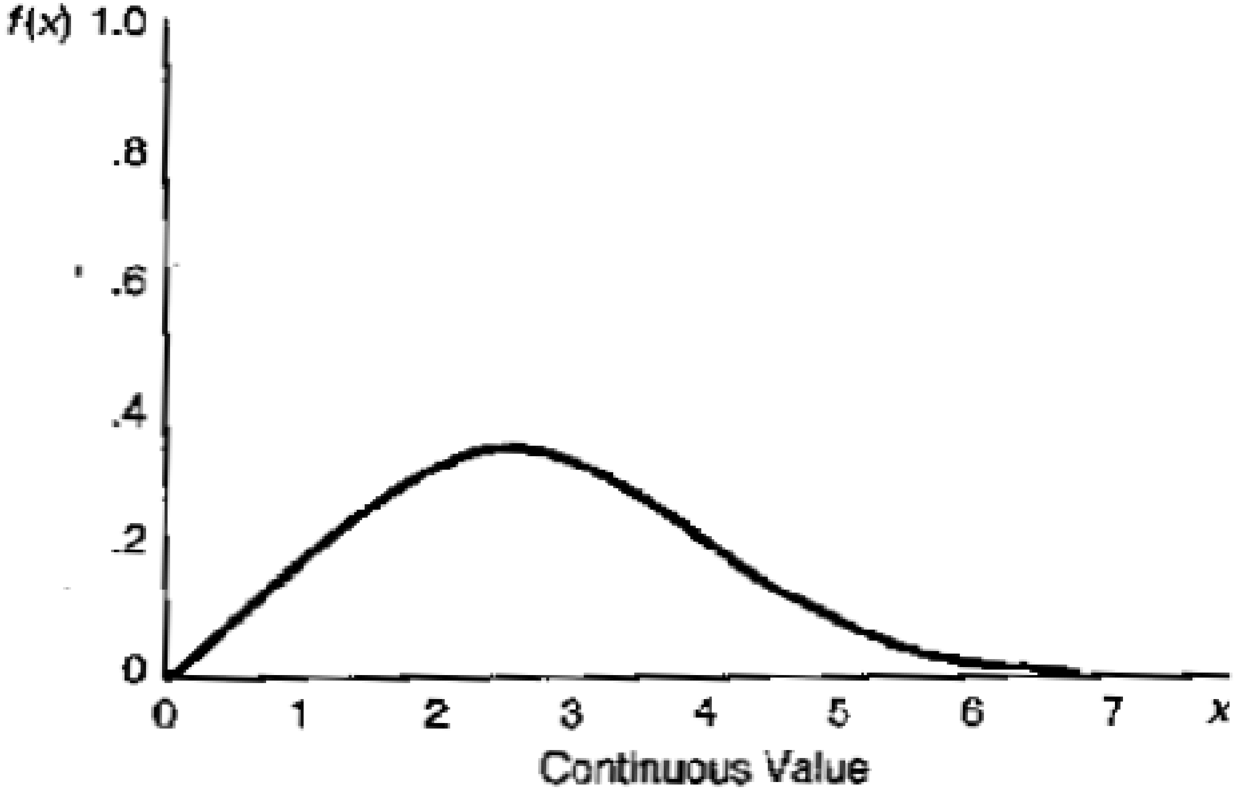
Random Behavior

An example of a discrete probability function



Random Behavior

An example of a continuous probability function



Simulating Random Behavior

- One of the most powerful features of simulation is its ability to mimic random behavior or variation that is characteristic of stochastic systems.
- Simulating random behavior requires that a method be provided to generate random numbers as well as for generating random variates based on a given probability distribution.

Generating Random Behavior

- Random behavior is imitated in simulation by using a random number generator.
- Random number generator is responsible for producing this stream of independent and uniformly distributed numbers.
- *The heart of the simulation is the generation of the random variates that drive the stochastic events in the simulation*

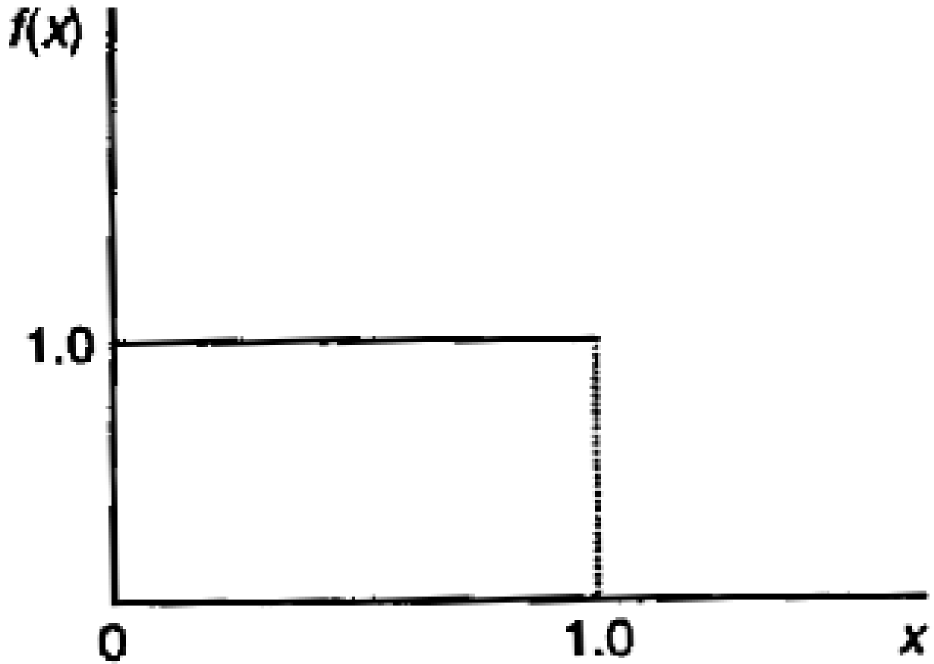
Generating Random Behavior

The U(0, 1) distribution of a random number generator

$$f(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{elsewhere} \end{cases}$$

$$\text{Mean} = \mu = \frac{1}{2}$$

$$\text{Variance} = \sigma^2 = \frac{1}{12}$$



Generating Random Behavior

- The numbers produced by a random number generator are not “random” in the truest sense.
  - The generator can produce the same sequence of numbers again and again, which is not indicative of random behavior.
  - They are often referred to as *pseudo-random number generators*.
- “Good” pseudo-random number generators can pump out long sequences of numbers that pass statistical tests for randomness (the numbers are independent and uniformly distributed).

Generating Random Behavior

- The most common method for generating random numbers is Linear Congruential Generators (LCG).
- Using LCG, a sequence of integers Z_1, Z_2, Z_3, ... is defined by the recursive formula:

$$Z_i = \left(aZ_{i-1} + c\right)\mathrm{mod}\left(m\right)$$

a : the multiplier;
c : the increment;
m : the modulus;
Z_0 : the seed or starting value;
a, c, m > 0 and integer

Linear Congruential Generators (LCG)

- The Z_i values are bounded by $0 \le Z_i \le m - 1$ and are uniformly distributed in the discrete case.
- The continuous version of the uniform distribution with values ranging between 0 and 1 can be obtained by

$$U_i = \frac{Z_i}{m}, \quad i = 1, 2, 3, \cdots$$

Linear Congruential Generators (LCG)

Example of LCG

$a = 21$
$c = 3$
$m = 16$

i	Z_i	U_i
0	13	
1	4	0.2500
2	7	0.4375
3	6	0.3750
4	1	0.0625
5	8	0.5000
6	11	0.6875
7	10	0.6250
8	5	0.3125
9	12	0.7500
10	15	0.9375
11	14	0.8750
12	9	0.5625
13	0	0.0000
14	3	0.1875
15	2	0.1250
16	13	0.8125
17	4	0.2500
18	7	0.4375
19	6	0.3750
20	1	0.0625

Linear Congruential Generators (LCG)

- The maximum cycle length that an LCG can achieve is *m*.
- To realize maximum cycle length, the values of *a*, *c*, and m have to be carefully selected.
- A guideline for the selection is:
  - $m = 2^b$ where *b* is determined based on the number of bits per word on the computer being used (for computer with 32 bits, $b = 31$)
  - *c* and *m* such that their greatest common factor is 1.
  - $a = 1 + 4k$, where *k* is an integer.

Linear Congruential Generators (LCG)

The LCG has full period if and only if the following three conditions hold (Hull and Dobell, 1962):

1. The only positive integer that (exactly) divides both m and c is 1
2. If q is a prime number (divisible by only itself and 1) that divides m, then q divides $a–1$
3. If 4 divides m, then 4 divides $a–1$

Linear Congruential Generators (LCG)

- Frequently, the long sequence of random number is subdivided into smaller segments referred to as *streams*.
- To subdivide the generator's sequence of random numbers into streams, it is need:
  - to decide how many random numbers to place in each stream
  - to generate the entire sequence of random numbers (cycle length) produced by the generator and record the Z_i values that mark the beginning of each stream.
- Each stream has its own starting or seed value.

Linear Congruential Generators (LCG)

- There are two types of LCG:
  - Mixed LCG
  - Multiplicative LCG
- Mixed LCG:
  - $c > 0$
- Multiplicative LCG
  - $c = 0$

Testing Random Number Generators

- The numbers produced by the random number generator must satisfy two properties:
  - Independent
  - Uniformly distributed between zero and one
- Generate a sequence of random numbers:

$$U_1, U_2, U_3, \ldots .$$

Testing Random Number Generators

- The hypothesis for testing the independence property:

  H_0 : U_i values from the generator are independent

  H_1 : U_i values from the generator are not independent

- The most common statistical method
  - the run test
  - the runs above and below the median test
  - the runs up and down test

Testing Random Number Generators

- The hypothesis for testing the uniformly property:

  H_0 : U_i values are U(0, 1)

  H_1 : U_i values are not U(0, 1)

- The most common statistical methods:
  - The Kolmogorov-Smirnov test
  - The chi-square test