Modeling & Simulation (SI-214)
Assignment#4

Modeling & Simulation (SI-214) Assignment#04

Table of Context

Written by: M Abdullah (192)

Source: Bard & ChatGPT

Q1: Describe Monte Carlo simulation method in detail and how Monte Carlo simulation is a special case of stochastic simulation?

Ans: Monte Carlo Simulation

Monte Carlo simulation is a statistical approach used to estimate the probability of multiple outcomes in a process that cannot be easily predicted due to the presence of random variables. To estimate numerical outcomes, statistical modeling and random sampling techniques are used. The Monte Carlo simulation method was named after Monaco's Monte Carlo Casino, which is famous for its games of chance.

Method for a Monte Carlo simulation

1) **Define the problem**: What are you trying to model or predict? What are the inputs to the model, and what are the possible outcomes?
2) **Define the probability distributions**: For each input variable, define a probability distribution that represents the range of possible values and their likelihood.
3) **Generate random samples**: Use a random number generator to generate random values for each input variable, according to the specified probability distributions.
4) **Run the simulation**: Determine the output of the model for each random observation.
5) **Analyze the results:** Calculating averages, variances, percentiles, or other relevant statistical measurements that may be required. These findings provide information about the system's behavior and dangers.
6) **Make Decisions:** Make decisions based on the simulation's results. Decision-makers can make better decisions by analyzing the risks involved and understanding the range of possible outcomes and their probability.

Monte Carlo Simulation as a Special Case of Stochastic Simulation

A stochastic simulation is a type of simulation that uses random variables to predict outcomes. Monte Carlo simulation is a special case of stochastic simulation that focuses on estimating numerical outcomes through random sampling. It is a flexible and effective technique used in various industries, such as finance, engineering, and project management, to make decisions in uncertain conditions. The terms "Monte Carlo" and "stochastic simulation" are sometimes used simultaneously, and Monte Carlo simulation is also a subset of stochastic simulation methods.

Q2: Explain any 5-blocks in GPSS?

Ans: GPSS

GPSS (General Purpose Simulation System) is a simulation language and software tool used to model and analyze complex systems in engineering, operations research, and computer science. GPSS uses blocks to construct simulation models. GPSS blocks specify how elements of a network behave and communicate with one another. The values provided in each block can be changed to simulate various scenarios. Each block represents a certain system process. The system output is analyzed to gain information that is useful for real-world systems.

Here are five major GPSS blocks

1) **GENERATE Block:**

The GENERATE block creates entities at specific points in time for representing the input of objects, events, or processes into the simulated system. It generates entities with specified attributes and schedules their arrival into the simulation system.

Syntax: GENERATE (A, T) where: A: The attribute value of the generated entity. T: The time delay until the next entity is generated.

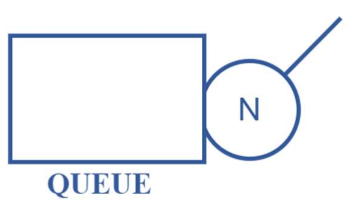
Symbolic Diagram:

2) **QUEUE Block:**

The QUEUE block represents a waiting line or queue where entities are held until they can proceed to the next process or block in the simulation. It maintains the queue discipline (e.g., First-In-First-Out, Last-In-First-Out) and holds entities until they may be processed.

Syntax: QUEUE N where: N is the name of queue

Symbolic Diagram:

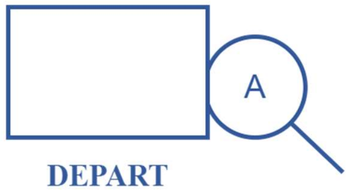


3) **DEPART Block:**

The DEPART block represents the exit point for entities leaving the system after completing their processing. It removes entities from the system, usually after they have been processed.

Syntax: DEPART A where: A is the name of the queue

Symbolic Diagram:



DEPART

4) **ADVANCE Block:**

The ADVANCE block represents the passage of time within the simulation. It adds a certain amount of time to the simulation clock.

Syntax: ADVANCE (T) Where: T: The time delay by which the simulation clock should advance.

Symbolic Diagram:

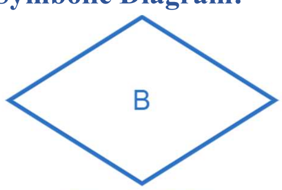
ADVANCE

5) **TRANSFER Block**

The TRANSFER block allows entities to switch between various processes or queues by moving them from one point to another in the system. It moves entities within the simulation from one block to another.

Syntax: TRANSFER (B) where: B: The block to which entities are transferred.

Symbolic Diagram:



TRANSFER

These are the five fundamental building blocks of GPSS for modeling and simulating various systems, with additional blocks and features for more complex modeling requirements, but these basic blocks are essential for understanding and constructing simulation models.

Q3: What are feedback systems? Write a short note on feedback system?

Ans: Feedback systems

Feedback systems are fundamental ideas in engineering and control theory, in which output affects input by either reinforcing it (positive feedback) or counteracting it (negative feedback). To control, stabilize, or amplify processes, feedback systems are essential in fields like electronics, biology, economics, and engineering. Feedback loops keep an eye on output and modify input to produce the desired results. Control systems, engineering, and economics all use feedback loops.

Example

The simplest example of a feedback system is thermostats, which monitor the room's temperature and alter the settings on the heater or air conditioner to maintain optimal conditions.

Components of a Feedback System

1. **Input:** The initial signal or stimulus applied to the system.
2. **Process (Plant):** The device or method used to change the input signal.
3. **Output:** The outcome or response that the system produces.
4. **Feedback Path:** The path through which some output is transmitted back into the system's input.
5. **Controller:** A component that decides how to generate an error signal by comparing the feedback output to the desired output.
6. **Error Signal:** The difference between the desired and actual outputs determined by the controller.
7. **Actuator:** Converts the controller's output (error signal) into a system-influencing control action.
8. **Sensor:** Tracks the system's output and informs the controller.
9. **Feedback Loop:** The closed loop created by the plant, controller, and feedback path.

Types of Feedback

- **Negative feedback** lowers the influence of disturbances on a system by comparing output to a desired setpoint and adjusting input to return it to that setpoint. **For instance,** a thermostat is a classic example of a negative feedback system. It measures the temperature of a room and adjusts the setting of a heater or air conditioner to maintain a desired temperature.

- **Positive feedback** increases the influence of disturbances by comparing output to a desired setpoint and adjusting the input to drive the output further away from the setpoint. **For instance,** social media platforms use positive feedback loops to keep users engaged. like, when you post something on social media, you receive likes and comments from your friends and followers. This positive feedback can encourage you to post more often, which keeps you coming back to the platform.

Q4: Explain the Markova chains with example and its application?

Ans: Markov Chains:

The Markov Chain is named after the Russian mathematician Andrey Markov. The probability of each random event in a Markov chain depends entirely on the state achieved by the previous event. Markov chains are stochastic processes. In other words, the system's future state depends completely on its existing state and not on how it achieved that state.

A Markov chain consists of:

1. **States:** These represent the different situations or conditions of the system.
2. **Transitions:** Transitions: These are the probabilities of going from one state to another. The probabilities, also known as transition probabilities, indicate the possibility that a state will change.

Example:

Consider a weather model with two states: "Sunny" and "Rainy." The transition probabilities might look like this:

- If it's sunny today, there is a 70% chance it will be sunny tomorrow and a 30% chance it will rain.
- If it's rainy today, there is a 60% chance it will be rainy again tomorrow and a 40% chance it will be sunny.

This can be represented as a transition matrix:

	Sunny	Rainy
Sunny	0.7	0.3
Rainy	0.4	0.6

Applications:

1) **Weather Forecasting:** Weather conditions often follow Markovian properties. Current weather conditions can be used to predict the probability of future weather conditions.
2) **Economics and Finance:** Markov chains are used in modeling stock prices, exchange rates, and other economic indicators. Investors use these models to make predictions about future market conditions.
3) **Game Theory:** Markov chains are used in various games to model players' strategies and outcomes.
4) **Genetics:** Markov models can be used to understand DNA sequences and the patterns within them.
5) **Queueing Theory:** Markov chains are applied to model systems where entities wait in line for service, such as customers in a queue at a supermarket checkout.
6) **Google's PageRank Algorithm:** PageRank, the algorithm Google uses to rank websites in its search engine results, can be modeled using a Markov chain.

Q5: Explain the properties of poisons process?

Ans: Poison Process

A poison process is a stochastic model that forecasts events or arrivals in a continuous-time environment. It is distinguished by exponentially distributed interarrival delays between successive events and a rate parameter λ. This process is characterized by two key properties:

- **Independence:** The occurrence of one event doesn't affect the probability of another event.
- **Constant Rate:** The average rate at which events occur remains constant over time or space.

Poison processes are used to simulate a variety of phenomena, including the arrival of customers at a business, the quantity of radioactive particles released by a substance, and the frequency of data transmission failures.

Examples of Poison processes:

- The number of cars that pass through an intersection in a given hour.
- The number of customers that arrive at a restaurant in a given day.
- The number of raindrops that fall on a given square meter in a given minute.
- The number of mutations that occur in a given DNA sequence during replication.

Properties of Poison processes:

1. **Stationarity:** The probability distribution of events occurring in a given time interval is the same as the probability distribution of events occurring in any other time interval of the same length. This demonstrates that the average arrival rate is constant during time.
2. **Independent increments:** The frequency of events in distinct time intervals is independent of each other, which means that events in one interval do not affect events in another. The total number of events in two distinct time intervals equals the individual count of each interval.
3. **Memory lessness:** The probability of an event occurring in a particular time interval is independent of the number of events that have previously occurred in that interval or any preceding interval, meaning that the process has no memory of its previous state.
4. **Poison distribution:** A statistical model that use the rate parameter λ and the interval's length to calculate the average number of events that will occur within a given period of time.

Poison processes are a theoretical model used in various disciplines (including physics, chemistry, engineering, biology, and economics) to model random phenomena, such as customer arrivals, phone calls, and vehicle arrivals. They are useful for approximating real-world systems, but finding a Poison process that perfectly models these phenomena can be challenging.

Q6: Write a short note on inverse transform techniques?

Ans: Inverse Transform Techniques

Inverse transform techniques are essential in probability theory and statistics, particularly in the generation of random variables. These strategies use a random variable's cumulative distribution function (CDF) to generate values that follow a certain probability distribution. The basic idea behind these strategies is to generate random variables with precise probability distributions, which increases statistical analysis accuracy and efficiency.

Working of inverse transform techniques

1. Understanding the CDF of a random variable X, denoted by $f(x)$ gives the probability that X takes on a value less than or equal to x. Mathematically,

   $f(x) = P(X \leq x)$

2. Generate a random number U from the uniform distribution on the interval [0, 1].
3. Find the inverse CDF of the desired distribution, $f^{-1}(U)$
4. Set $X = f^{-1}(U)$ The random variable X will have the desired distribution.

Applications of inverse transform techniques

1. **Monte Carlo simulations**: Inverse transform techniques are utilized in Monte Carlo simulations to generate random variables for modeling complex systems in engineering, economics, and other scientific disciplines, enabling estimation and prediction of results.
2. **Queuing theory:** Inverse transform techniques are used in queueing theory to analyze and optimize queueing systems in real-world settings such as telecommunications, computer networks, and service systems.
3. **Healthcare Systems:** In healthcare systems, inverse transform techniques can be used to model patient arrival and service times, allowing for better resource utilization and patient care.
4. **Environmental studies:** Inverse transform methods used to predict weather emergencies such as floods or droughts and produce random variables to assess the effects on the ecosystem.

Inverse transform techniques are essential in fields like statistics, machine learning, and modeling for creating random variables without a closed-form inverse CDF (a closed-form inverse CDF is an expression that calculates the value of x such that F(x) = p, where F is the distribution's CDF and p is a probability value). They utilize cumulative distribution functions and uniform random variables to generate specific distributions, enhancing statistical analysis accuracy and efficiency, and are utilized in Monte Carlo simulations, queueing theory, healthcare systems, and environmental studies.

Q7: Explain co-variance and co-relation?

Ans: Covariance

Covariance is a measure of how two variables change together. If the covariance is positive, the variables increase or decrease together. If it's negative, one variable increases when the other decreases.

Formula for Covariance (for a sample):

$$\mathbf{Cov(X,Y) = \frac{\sum(x_i - \overline{X})\,(y_i - \overline{Y})}{n-1}}$$

Where:

- X and Y are the two sets of variables.
- $Cov(X,Y)$ mean covariance between X and Y.
- x_i and y_i are individual data points for X and Y respectively.
- $\bar{X}$ and $\bar{Y}$ are the means of X and Y, respectively.
- n is the number of data points.

Basic Properties of Covariance:

1. **Positive Covariance:** Indicates a positive relationship between variables.
2. **Negative Covariance:** Indicates a negative relationship between variables.
3. **Zero Covariance:** Indicates no linear relationship between variables.

Correlation

Correlation is a standardized measure of the strength and direction of a linear relationship between two variables, with a value between -1 and 1. 1 indicates a perfect positive, -1 indicates a perfect negative, and 0 indicates no correlation.

Formula for Correlation (Pearson correlation coefficient):

$$\mathbf{Correlation(X,Y) = \frac{Cov(X,Y)}{\sigma_{X\times}\sigma_Y}}$$

Where:

- X and Y are the two sets of variables.
- $Correlation(X,Y)$ mean correlation between X and Y.
- $Cov(X,Y)$ mean covariance between X and Y.
- σ_X and σ_Y are the standard deviations of X and Y respectively.
- General Formula for σ_T

$$\sigma_T = \frac{\sum(t_i - \overline{T})^2}{n}$$

where: T is a set of variable and T_i is a individual data points in set T.

Basic Properties of Correlation:

1. **Range:** Correlation values range from -1 to 1.
2. **Magnitude:** The stronger the relationship, the closer the correlation is to 1 or -1.
3. **Direction:** Positive correlation (near to 1) indicates a positive relationship. Negative correlation (near to -1) indicates a negative relationship.

Applications of covariance and correlation

Covariance and correlation are used in a variety of fields such as statistics, finance, and economics to assess investment risk and analyze economic variables such as GDP and unemployment. Here are some examples of how covariance and correlation can be used:

- A **financial analyst** might use covariance to assess the risk of a stock portfolio.
- A **marketing analyst** might use correlation to assess the link between customer expenditure and advertising spending.
- A **medical researcher** may utilize correlation to assess the association between a new medicine and the symptoms of a patient.

Q8: What is meant by clock time in simulation? How do you update the clock time in simulation?

Ans: Clock Time in Simulation

The clock time in a simulation refers to the virtual time that passes inside the simulation environment, measuring the progress of events and activities. Clock time is used to organize and coordinate multiple tasks, processes, and events inside the simulated system, ensuring accurate tracking and control. The clock time is commonly measured in seconds, minutes, or hours.

Update the Clock Time in Simulation

In a simulation, updating the clock time involves sending the simulation time from its present state to a future one, either in discrete or continuous steps. The simulator keeps track of the current simulation time, which is updated at each simulation step and might be constant or variable depending on the simulation nature. There are two main ways to update the clock time in simulation:

1. **Discrete Event Simulation (DES):** The clock time jumps from one event to the next in discrete event simulation, indicating important system events are complete, such as client arrival or task completion. In discrete event simulation (DES), the simulator updates the clock time to the next scheduled event, expressing modifications to the nature of the simulated system. To update the clock time in a DES simulation, the simulator typically follows these steps:
   - **i.** Identify the next scheduled event.
   - **ii.** Update the clock time to the time of the next scheduled event.
   - **iii.** Execute the next scheduled event.
   - **iv.** Repeat steps i to iii until the simulation is complete.
2. **Continuous Simulation (CTS):** In continuous simulation, time is treated as a continuous variable, and changes in system variables are calculated over small intervals of time. It updates the simulation time in short stages, making it suited for systems that are continually changing, such as real-world ones. To update the clock time in a CTS simulation, the simulator typically follows these steps:
   - **i.** Calculate the change in the system state since the last simulation step.
   - **ii.** Update the clock time by the time required to calculate the change in the system state.
   - **iii.** The system state should be updated to indicate the status change since the last simulation step.
   - **iv.** Repeat steps i to iii until the simulation is complete.

The method used to update the clock time in a simulation is determined by the type of simulation and the level of accuracy desired. CTS is used for continuous systems such as physical or chemical systems, whereas DES is utilized for discrete systems such as computer networks or manufacturing. Simulation software and modeling techniques were also used to determine the method. The accuracy with which the simulation time is updated is essential for the reliability and validity of the results because it ensures that the simulation appropriately represents the behavior of the real-world system that is being modeled.

Example

A DES simulation of a traffic light involves tracking the current simulation time and a list of scheduled events. The simulator identifies the next scheduled event, advances the clock time to the next time, and executes the event. This process repeats until the simulation is complete, ensuring that the traffic light changes from green to yellow. The simulator maintains a variable to track the current simulation time and lists of scheduled events.

Q9: Describe different types of statements and statement used in CSMP with suitable examples?

Ans: CSMP

CSMP stand for Continuous System Modeling Program. CSMP was a programming language developed in the 1960s for simulating continuous systems. There are three main types of statements in CSMP:

1) **Structural statements** define the model structure, such as the variables, parameters, and equations in the system.

Example

- **PARAMETER:** This statement assigns a value to a parameter, e.g.
```
PARAMETER K = 1.0;
```
This statement assigns the value 1.0 to the parameter K.
- **INTEGRATE:** This statement defines a differential equation, e.g.
```
INTEGRATE X = K * X;
```
This statement defines the differential equation $\frac{dx}{dt} = K \times X$.
- **INITIAL:** This statement specifies an initial condition for a variable, e.g.
```
INITIAL X = 1.0;
```
This statement specifies that the initial value of the variable X is 1.0.

2) **Data statements** assign values to parameters and initial conditions.

Example

- **INCON:** This statement specifies the initial value of an integration function block, e.g.
```
INCON X = 1.0;
```
This statement specifies that the initial value of the integration function block X is 1.0.
- **DATA:** This statement assigns a value to a data variable, e.g.
```
DATA R = 1.0;
```
This statement assigns the value 1.0 to the data variable R.

3) **Control statements** specify options for assembling and executing the model, such as the integration method and output format.

Example

- **PRINT:** This statement prints the value of a variable to the output file, e.g.
```
PRINT X;
```
This statement prints the value of the variable X to the output file.
- **PRDEL:** This statement prints the derivative of a variable to the output file, e.g.
```
PRDEL X;
```
This statement prints the derivative of the variable X to the output file.
- **STOP:** This statement terminates the simulation, e.g.
```
STOP AFTER 10.0;
```
This statement terminates the simulation after 10.0 seconds.

Q10: Perform the simulation of the following inventory system is given daily demand is represented by random numbers 4, 3,8,2,5 and the demand probability are given by:

DEMAND	PROBABILITY
0	0.2
1	0.5
2	0.3

If the initial inventory is 4 units determine on which day the shortest conditions occur?

Ans: We use the Monte Carlo simulation method for this numerical problem.
Steps for a Monte Carlo simulation

1) **Define the problem**
Given:

Simulating Inventory System with:
Random Numbers: 4, 3, 8, 2, 5. that represent daily demand
Initial Inventory: 4 units
Demand Probability Table

DEMAND	PROBABILITY
0	0.2
1	0.5
2	0.3

To Find:

On which day the shortest conditions occur?

2) **Probability Distributions**
Important Information

The random-digit interval is determined by how many digits are in the given random numbers, and each interval ending is determined by probability, like: In our case, there are only one-digit random numbers, which start at 0 and end at 9, and these are 10 in counting. So

0.2 numbers out of 10 are 2 — 1st interval has the first two numbers,
0.5 numbers out of 10 are 5 — 2nd interval has the next five numbers,
0.3 numbers out of 10 are 3. — 3rd interval has the next 3 numbers

Table:A			
Demand	**Probability**	**Cumulative Probability**	**Random Digits Interval**
0	0.2	0.2	0-1
1	0.5	0.7	2-6
2	0.3	1.0	7-9

3) **Random Samples**

Random numbers are given, which are: 4, 3, 8, 2, 5.

4) Run the simulation

In Given, the random numbers represent the daily demand, so there are 5 random numbers, and then there are 5 days in the simulation as well.

Important Information

Day Demand: In Table:A, check in which interval the random number lies and write its corresponding demand.

Remaining Inventory: Initial Inventory – Day Demand

Day	Random Numbers for Demand	Initial Inventory	Day Demand	Remaining Inventory
1	4	4	1	3
2	3	3	1	2
3	8	2	2	0
4	5	0	1	-1

5) Analyzing the results

Day	Shortest Condition
1	0
2	0
3	0
4	-1

On day 4, the inventory became out of stock, so on day 5, the inventory was also out of stock. That's why the simulation stopped on day 4.

6) Decisions

On day 4 the shortest condition occurs if initial inventory is 4 units