Input Modeling and Output Analysis

1 Input Modeling

Input modeling is a critical component of the modeling and simulation process because it ensures that the simulation accurately reflects the real-world system being studied. There are several reasons why it is essential:

1. **Realism:** Proper input modeling ensures that the simulation inputs accurately represent the variability and characteristics of the real-world system.
2. **Predictive Accuracy**: Well-modeled inputs enable the simulation to make reliable predictions about the system's future performance under various conditions. This is especially important for decision-making and planning.
3. **Validation and Verification**: Input modeling helps in validating the simulation model by ensuring that the inputs reflect real-world data. A validated model is more likely to produce accurate and trustworthy results.
4. **Sensitivity Analysis**: By accurately modeling inputs, sensitivity analysis can be performed to understand how changes in input variables affect the output of the simulation. This helps in identifying critical inputs that significantly influence the system's behavior.

In general, for the **stochastic process** of the inputs, we care whether it is **autonomous**, **stationary** and **homogeneous**. For now, we assume the input process is autonomous.

1.1 Stationarity

A process is considered stationary if its **statistical properties**, such as **mean**, **variance** and **autocorrrelation** are constant over time. In math, we say $\{X_{t_1}, X_{t_2}, \cdots, X_{t_r}\}$ and $\{X_{t_1+s}, X_{t_2+s}, \cdots, X_{t_r+s}\}$ have the same distribution for any s. Notice that **Stationarity** implies identical distributioon but not independence. For example:

$$Z_t = Y_t + X \tag{1}$$

Z_t are identically distributed but all depend on X. To assess input's stationarity and independence, we could calculate the **sample mean**:

$$\bar{x} \equiv \frac{1}{n} \sum_{t=0}^{T-1} x_t \tag{2}$$

And **sample autocovariance**, where h is the time lag:

$$\hat{\gamma}(h) \equiv \frac{1}{n} \sum_{t=0}^{n-|h|-1} (x_{t+|h|} - \bar{x})(x_t - \bar{x}) \tag{3}$$

And **autocorrelation** for independence:

$$\hat{\rho}(h) \equiv \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \tag{4}$$

1.2 Shape of Distribution

To determine the shape of distribution, we could use histogram to visualize the data:

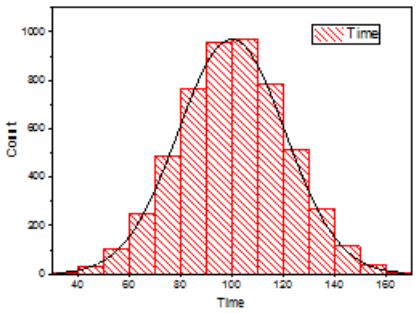

Figure 1: Histogram

The basic idea of histogram is to discretize the data, and divide the range of observed values into bins. Then, we count number of values that fall into each bin and plot. Usually, if we have n data points, we will use $n^{1/2}$ or $(2n)^{1/3}$ bins.

2 Output Analysis

Similar with input modeling, output analysis is useful for Validation and Verification, Performance Evaluation, Uncertainty Quantification and Optimization.

2.1 Bounded-Horizon Study

A bounded horizon study in output analysis refers to evaluating the performance of a simulation model over a **finite period or a limited number of time steps**. In this case, we care more about the performance metrics, such as the sample mean. To assess whether the sample mean is reasonable, we could utilize **Central Limit Theorem**. Assume the output as a random variable $[Y_i]$, then the expected mean and variance will be:

$$E[Y_i] = \mu \tag{5}$$

$$Var[Y_i] = E[(Y_i - \mu)^2] = \sigma^2 \tag{6}$$

When n is large enough, the sample mean will approximate to normal distribution:

$$\bar{Y}(n) = \frac{1}{n}\sum_{i=1}^{n\to\infty} Y_i \approx N(\mu, \frac{\sigma^2}{\sqrt{n}}) \tag{7}$$

Now we assume the simulation outputs are $y_1, y_2, y_3, \cdots, y_n$, then the actual sample mean and variance will be:

$$\bar{y}_n = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{8}$$

$$s_n^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_n)^2 \tag{9}$$

For 90% confidence interval ($\alpha = 0.05$) and $n = 10$, we have:

$$t = t(9, 0.95) \tag{10}$$

Then the confidence interval will be:

$$\bar{y}_n \pm t\sqrt{\frac{s_n^2}{n-1}} \tag{11}$$

2.2 Steady State Study

In output analysis, a steady-state study focuses on examining the long-term behavior of a system. The goal is to understand the system's performance metrics once it has reached equilibrium, where the effects of initial conditions are no longer significant, and the system's properties are relatively stable over time.

In general, we care more about **warm-up period**, which is the transient period from initialization until simulated system reaches steady state. To find the warm-up period, we could use **Welch's method** to define such a period. The procedures include:

1. Divide the simulation output data:
    - Run the simulation and collect the output data of interest.
    - Divide the output data into multiple, possibly overlapping, segments.
2. Calculate Averages for Each Segment:
    - For each segment, compute the average value of the output data. This creates a time series of averages representing the system's behavior over time.
3. Plot the Segment Averages:
    - Plot the segment averages against time. This plot will help visualize the convergence of the system towards steady state.

- Alternatively, plot the cumulative average of the data over time to see how the overall average stabilizes.

4. Identify the Warm-Up Period:

   - Analyze the plot to determine the point at which the averages stabilize or show less variation. This point indicates the end of the warm-up period.
   - The data before this point can be considered part of the transient phase and should be discarded for steady-state analysis.

3 Verification and Validation

3.1 Verification

Verification involves checking that the model has been implemented correctly and accurately represents the designer's conceptual description and specifications. It ensures that the model is free from logical errors and functions as intended. (**Did I build the model correctly?**)

3.2 Validation

Validation, on the other hand, involves ensuring that the model accurately represents the real-world system it is intended to simulate. It ensures that the model's outputs are accurate and can be trusted for decision-making purposes. (**Did I build the correct model?**)