ACI650 - Modelos y Simulación

Verification and Validation of Simulation Models

Mario González
Facultad de Ingeniería y Ciencias Ambientales
Centro de Investigación, Estudios y Desarrollo de Ingeniería (CIEDI)

November 28, 2016

Learning Objectives

- Definitions of validation and verification
- Techniques for verification of simulation models
- Techniques for validation of simulation models
- Statistical Methods for Comparing real-world observations with simulation output data

M&S process (Recall)

- Understand the system under study
- Identify and formulate the problem
- Collect and process real system data
- Implement the model (and **verify**)
- **Validate** the model
- Perform Simulation
- Interpret the results
- Recommend course of action

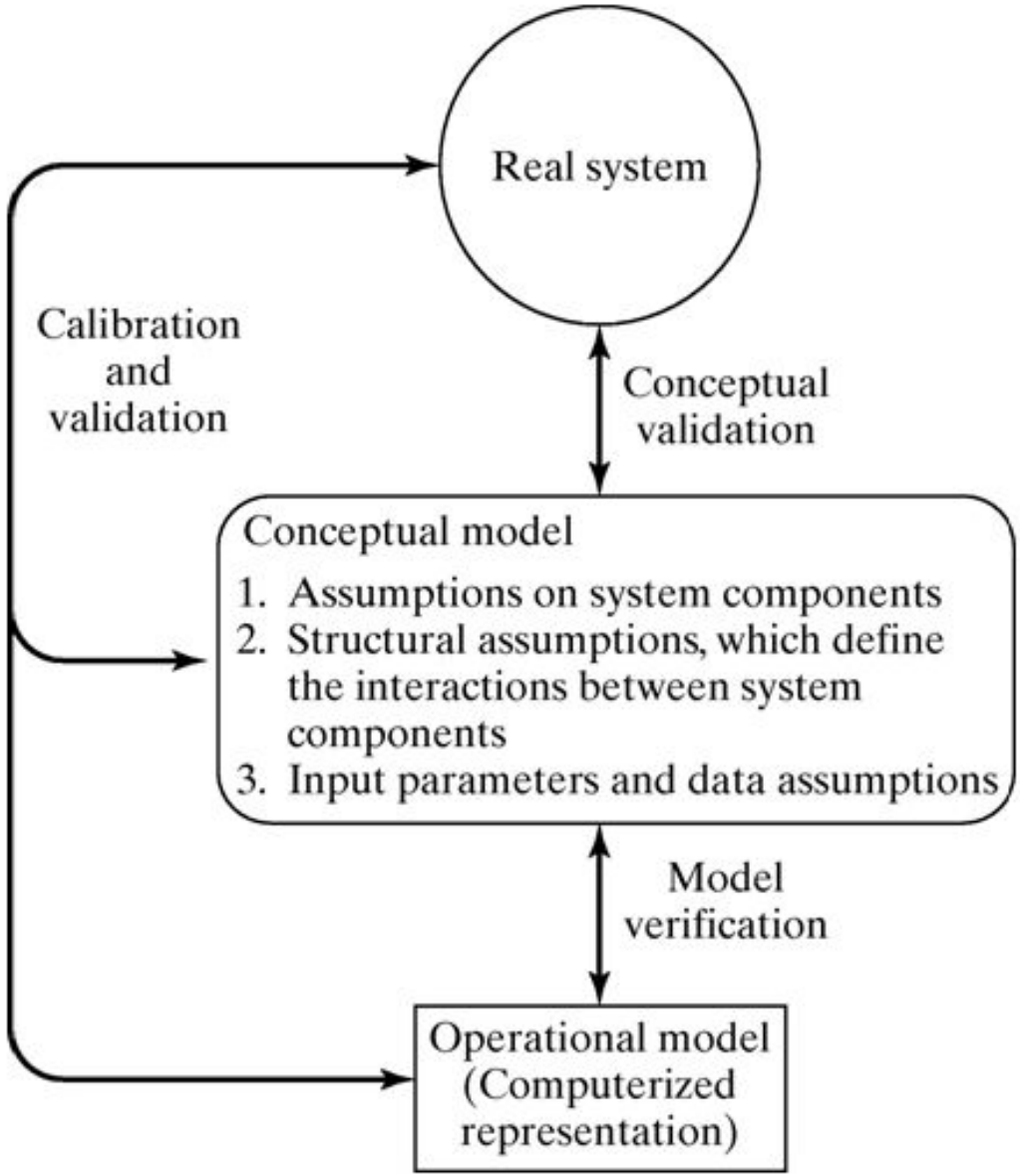
Introduction

- One of the most difficult problems facing the simulation analyst is determining whether a simulation model is an accurate representation of the actual system being studied (i.e., whether the model is valid).
- If the simulation model is not valid, then any conclusions derived from it is of virtually no value.
- Validation and verification are two of the most important steps in any simulation project.

Verification and Validation

- **Verification:** is concerned with building the **model right**.
  - It is utilized in the comparison of the **conceptual model** to the **computer representation** that implements that conception.
  - It asks the questions: Is the model implemented correctly in the computer? Are the input parameters and logical structure of the model correctly represented?
- **Validation:** is concerned with building the **right model**.
  - It is utilized to determine that a model is an **accurate representation** of the real system.
  - Validation is usually achieved through the **calibration of the model**, an **iterative process** of comparing the model to actual system behavior and using the discrepancies between the two, and the insights gained, **to improve the model**.
  - This process is repeated until model accuracy is judged to be acceptable.

Verification and Validation



Verification

- **Purpose:** ensure the conceptual model is reflected accurately in the computerized representation.
- Many common-sense suggestions, for example:
  - Have someone else check the model.
  - Make a flow diagram that includes each logically possible action a system can take when an event occurs.
  - Closely examine the model output for reasonableness under a variety of input parameter settings.
  - Print the input parameters at the end of the simulation, make sure they have not been changed inadvertently.
  - Make the operational model as self-documenting as possible.
  - If the operational model is animated, verify that what is seen in the animation imitates the actual system.
  - Use the debugger.
  - If possible use a graphical representation of the model.

Verification from an implementation POV I

- General good programming practice
  - Once the simulation model has been programmed, the analysts/programmers must check if this computer code contains any programming errors (bugs).
  - Software engineers have developed numerous procedures for writing good computer programs and for verifying the resulting software, in general (not specifically in simulation).
  - A few key terms are: modular programming, object oriented programming, chief programmer's approach, structured walkthroughs, correctness proofs.

Verification from an implementation POV II

- Verification of intermediate simulation output
  - The analysts may calculate some intermediate simulation results manually, and compare these results with outputs of the simulation program.
  - Getting all intermediate results from a computer program automatically is called **tracing**.
  - The analyst may check modules such as:
    1. The analysts may test the pseudorandom number generator separately, if they had to program that generator themselves or they do not trust the software supplier's expertise.
    2. The analysts may further test the subroutines that generate samples from certain non-uniform distributions. Beware of the lack of standard notation: some authors and some software use the notation $N(\mu, \sigma)$, whereas others use $N(\mu, \sigma^2)$. Also, be careful with the unit of measures.

Verification from an implementation POV III

- Comparing final simulation outputs with analytical results
  - The analysts may verify the simulation response by running a simplified version of the simulation program with a known analytical solution.
  - Then the analysts can use a textbook on queueing theory to find formulas for the steady state expectations of several types of response (mean waiting time of jobs and mean utilizations of machines).
  - First the analysts can run the simulation program with exponential arrival and service times, only to verify the correctness of the computer program.
  - Next they run the simulation program with non-exponential input variables to simulate the responses that are of real interest to the users.

Verification from an implementation POV IV

- Statistical technique
  - The analyst can use a statistical test to verify that the expected value of y, the simulation response of the simplified simulation program, is equal to the known steady state mean μ_y.
  - The well-known Student t test which assumes normally and independently distributed simulation responses y with mean μ_y and variance σ_y^2, can be used to test the hypothesis $H_0 : E(y) = \mu_y$.

Verification from an implementation POV V

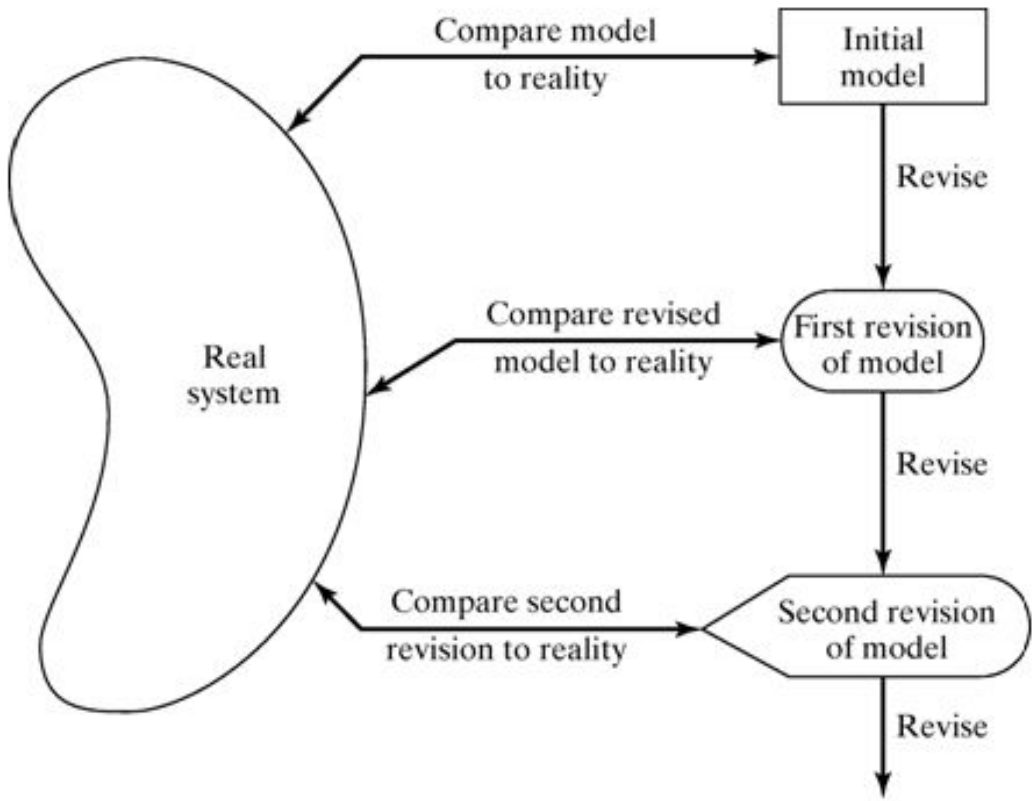
- Animation
  - To verify the computer program of a dynamic system, the analysts may use animation.
  - Since the users are familiar with the corresponding real system, they can detect programming errors.
  - Well-known examples are simulations that show how vehicles defy the laws of nature and cross through each other, and simulations that have customers who miraculously disappear during the simulation run.
  - Most simulation researchers agree that animation may be dangerous too, as the analysts and users tend to concentrate on very short simulation runs so the problems that occur only in long runs go unnoticed.

Validation

- Once the analysts believe that the simulation model is programmed correctly, they must face the next question: is the conceptual simulation model (as opposed to the computer program) an accurate representation of the system under study?

Calibration and Validation I

- **Validation:** the overall process of comparing the model and its behavior to the real system.
- **Calibration:** the iterative process of comparing the model to the real system and making adjustments.



Calibration and Validation II

- Comparison of the model to real system
  - **Subjective tests:** Ask to people who are knowledgeable about the system
  - **Objective tests:** Requires data on the real system's behavior and the output of the model
- Danger during the calibration phase
  - Typically few data sets are available, in the worst case only one, and the model is only validated for these.
  - Solution: If possible collect new data sets.
- No model is ever a perfect representation of the system
  - The modeler must weigh the possible, but not guaranteed, increase in model accuracy versus the cost of increased validation effort.

Calibration and Validation III

- Three-step approach for validation:
  1. Build a model that has high face validity.
  2. Validate model assumptions.
  3. Compare the model input-output transformations with the real system's data.

High Face Validity

- The objective of this step is to develop a model that, on the surface, seems reasonable to people who are familiar with the system under study.
- This step can be achieved through discussions with system experts, observing the system, or the use of intuition.
- It is important for the modeler to interact with the client on a regular basis throughout the process.
- It is important for the modeler to perform a structured walk-through of the conceptual model before key people to ensure the correctness of model's assumptions.

Validate Model Assumptions I

- General classes of model assumptions:
  - **Structural assumptions:** how the system operates.
  - **Data assumptions:** reliability of data and its statistical analysis.
- Bank example: customer queuing and service facility in a bank.
  - Structural assumptions
    - Customer waiting in one line versus many lines
    - Customers are served according FCFS versus priority
  - Data assumptions, e.g., interarrival time of customers, service times for commercial accounts.
    - Verify data reliability with bank managers
    - Test correlation and goodness of fit for data

Validate Model Assumptions II

- Test the assumptions of the model empirically:
  - In this step, the assumptions made in the initial stages of model development are tested quantitatively.
  - For example, if a theoretical distribution has been fitted to some observed data, graphical methods and goodness of fit tests are used to test the adequacy of the fit.
- **Sensitivity analysis** can be used to determine if the output of the model significantly changes when an input distribution or when the value of an input variable is changed.
  - Example: In most queuing systems, if the arrival rate of customers were to increase, it would be expected that server utilization, queue length and delays would tend to increase.
  - For large-scale simulation models, there are many input variables and thus possibly many sensitivity tests.
  - Sometimes not possible to perform all of theses tests, select the most critical ones.

Simulation output data representativeness

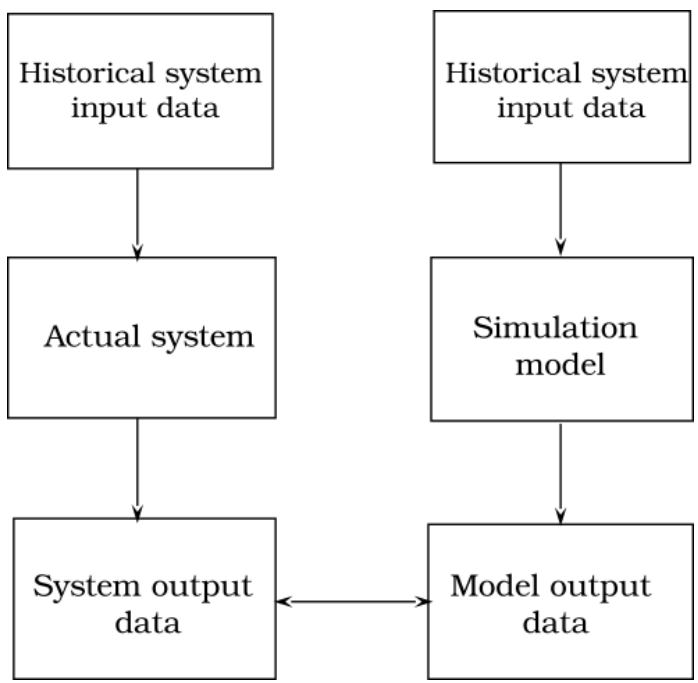
- The most definitive test of a model's validity is determining how closely the simulation output resembles the output from the real system.
- The Turing test can be used to compare the simulation output with the output from the real system.
  - The output data from the simulation can be presented to people knowledgeable about the system in the same exact format as the system data.
  - If the experts can differentiate between the simulation and the system outputs, their explanation of how they did that should improve the model.
- Statistical methods are available for comparing the output from the simulation model with those from the real-world system.

Comparing the real-world system and simulation outputs I

- Two approaches for comparing the outputs from the real-world system with the simulation outputs are:
  - Inspection Approach
  - Confidence-Interval Approach

Comparing the real-world system and simulation outputs II

- Inspection Approach
  - The output data of the real system and the simulated system can be plotted such that the horizontal axis denotes time and the vertical axis denotes the real and simulated values respectively.
  - The users may eyeball timepaths to decide whether the simulation model “accurately” reflects the phenomena of interest.

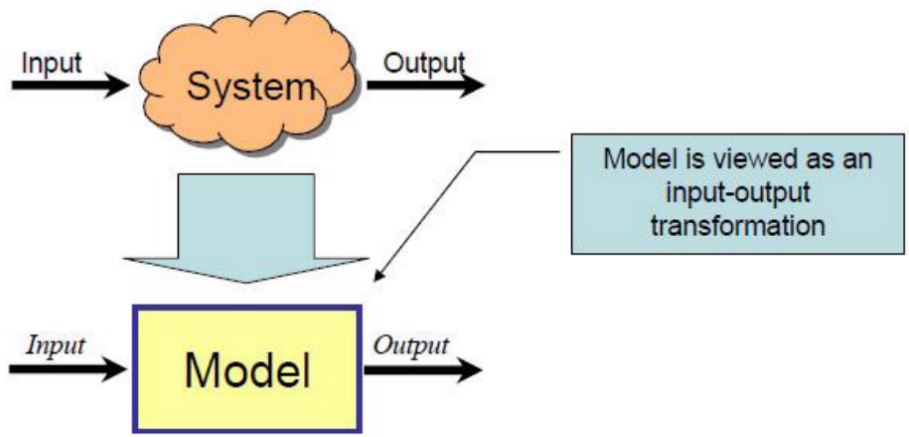


Comparing the real-world system and simulation outputs III

- Inspection approach (also called trace driven method) may provide valuable insight into the adequacy of the simulation model for some simulation studies.
- In fact, this may be the only feasible approach because of severe limitations on the amount of data available on the operation of some real-world systems.
- Care must, however, be taken in interpreting the result of this approach.

Validating Input-Output Transformation I

- **Goal:** Validate the model's ability to predict future behavior
  - The structure of the model should be accurate enough to make good predictions for the range of input data sets of interest.
- One possible approach: use historical data that have been reserved for validation purposes only.
- Criteria: use the main responses of interest.



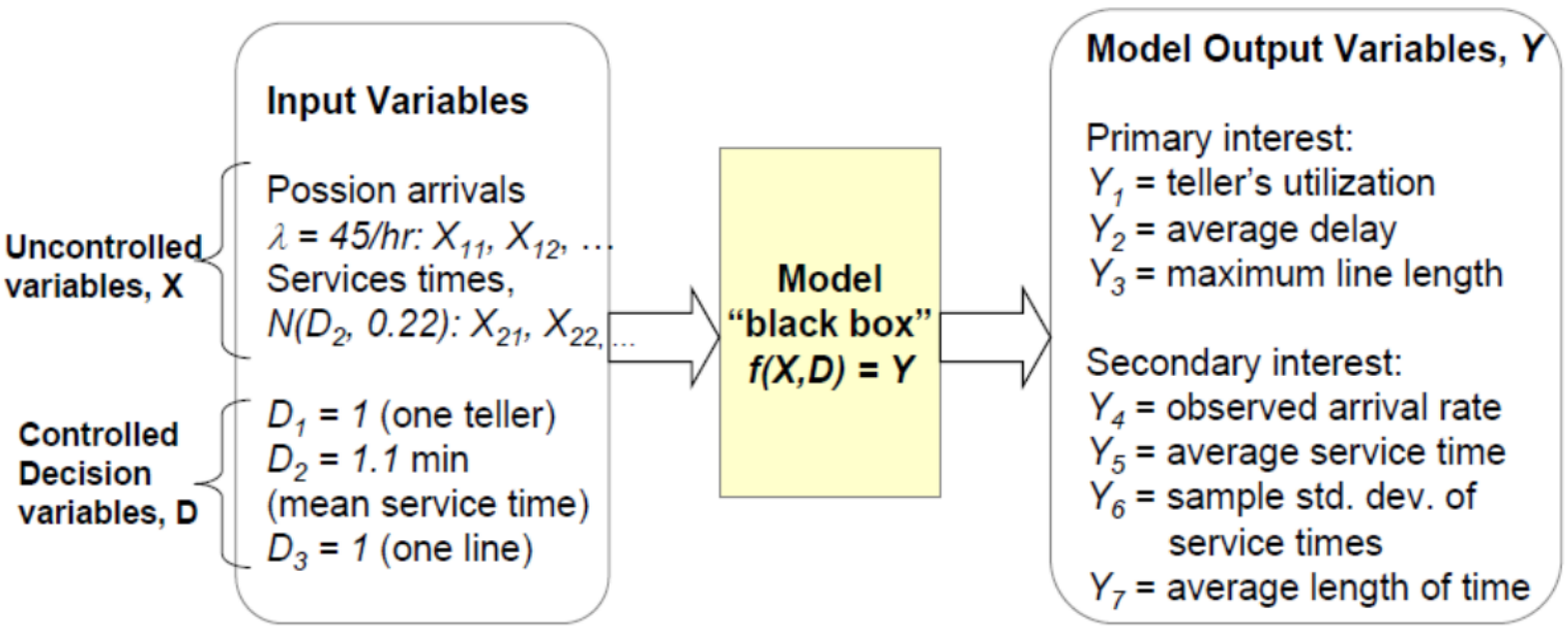
Validating Input-Output Transformation II

Bank Example

- **Data collection:** 90 customers during 11 am to 1 pm.
  - Observed service times $\{S_i, i = 1, 2, \ldots, 90\}$.
  - Observed interarrival times $\{A_i, i = 1, 2, \ldots, 90\}$.
  - Data analysis let to the conclusion that:
    - Interarrival times: exp. distr. with rate $\lambda = 45$
    - Service times: $N(1.1, 0.22)$

Validating Input-Output Transformation III

- A model was developed in close consultation with bank management and employees.
- Model assumptions were validated
- Resulting model is now viewed as a “black box”:



Comparison with Real System Data I

- Real system data are necessary for validation.
  - System responses should have been collected during the same time period (from 11am to 1pm on the same day.)
- Compare the average delay from the model Y_2 with the actual delay Z_2:
  - Average delay observed, $Z_2 = 4.3$ minutes, consider this to be the true mean value $\mu_0 = 4.3$.
  - When the model is run with generated random variates X_{1n} and X_{2n}, Y_2 should be close to Z_2.

Comparison with Real System Data II

- Six statistically independent replications of the model, each of 2-hour duration, are run.

Replication	Y_4 Arrivals/Hour	Y_5 Service Time [Minutes]	Y_2 Average Delay [Minutes]
1	51	1.07	2.79
2	40	1.12	1.12
3	45.5	1.06	2.24
4	50.5	1.10	3.45
5	53	1.09	3.13
6	49	1.07	2.38
Sample mean			2.51
Standard deviation			0.82

Hypothesis Testing I

- Compare the average delay from the model Y_2 with the actual delay Z_2
- Null hypothesis testing: evaluate whether the simulation and the real system are the same (with respect to the output measures):

$$H_0 : E(Y_2) = 4.3 \text{ min}$$

$$H_1 : E(Y_2) \neq 4.3 \text{ min}$$

- If H_0 is not rejected, then, there is no reason to consider the model invalid
- If H_0 is rejected, the current version of the model is rejected, and the modeler needs to improve the model

Hypothesis Testing II

- **Conduct the t-test:** Chose level of significance ($\alpha = 0.05$) and sample size ($n = 6$).
- Compute the same mean $\bar{Y}_2$ and sample standard deviation S_Y over the n replications.
- Compute test statistics (two-sided test):

$$t_0 = \left| \frac{Y_2 - \mu_0}{S/\sqrt{n}} \right| = \left| \frac{2.51 - 4.3}{0.82\sqrt{6}} \right| = 5.36 > t_{\alpha/2, n-1} = 2.5706$$

- Reject H_0. Conclude that the model is inadequate.

Type II Error I

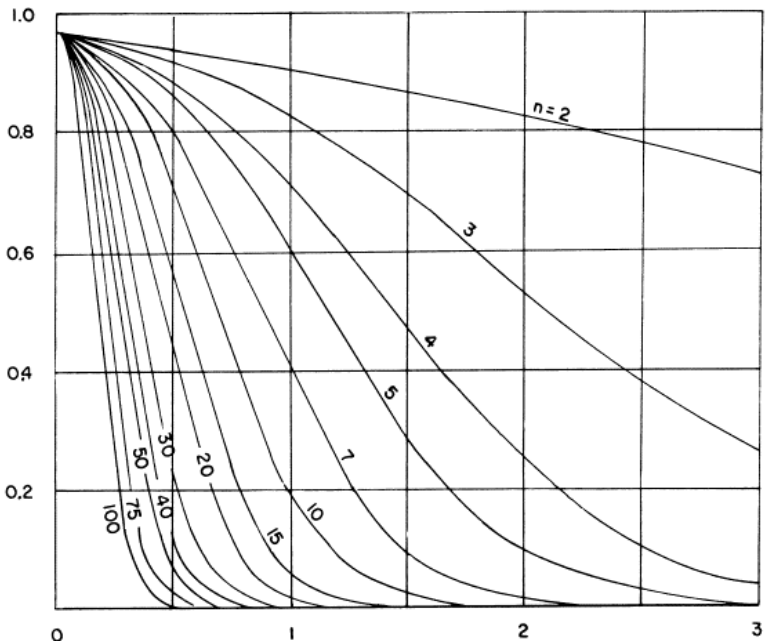
- For validation, the power of the test is:
  - Probability[detecting an invalid model] = $1 - \beta$
  - $\beta = P$ (Type II error) = P(failing to reject $H_0 | H_1$ is true)
  - Consider failure to reject H_0 as a strong conclusion, the modeler would want β to be small. Value of β depends on:
    - Sample size, n
    - The true difference, δ, between $E(Y)$ and μ:

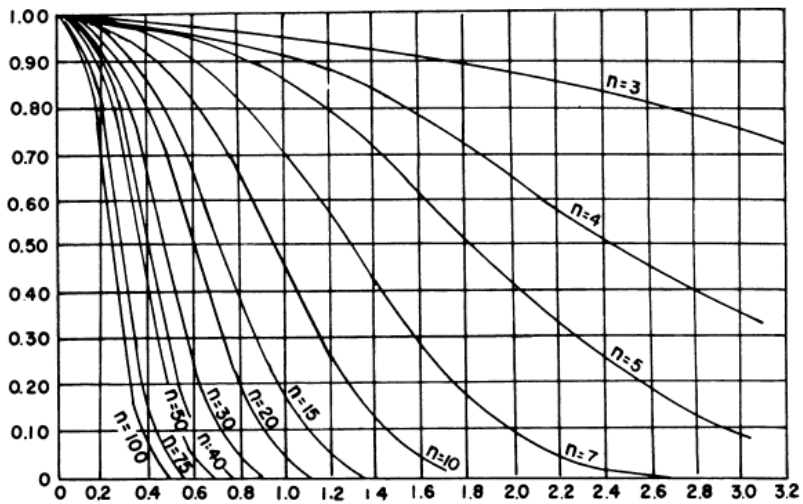
$$\delta = \frac{|E(Y) - \mu|}{\sigma}$$

- In general, the best approach to control β error is:
  - Specify the critical difference, δ.
  - Choose a sample size, n, by making use of the operating characteristics curve (OC curve).

Type II Error II

- Operating characteristics curve (OC curve) is the graph of the probability of a type II error $\beta(\delta)$ versus δ for a given sample size n





Type I and II Error

- Type I error (α):
  - Error of rejecting a valid model.
  - Controlled by specifying a small level of significance α.
- Type II error (β):
  - Error of accepting a model as valid when it is invalid. Controlled by specifying critical difference and find the value of n.
- For a fixed sample size n, increasing α will decrease β.

Confidence Interval Testing I

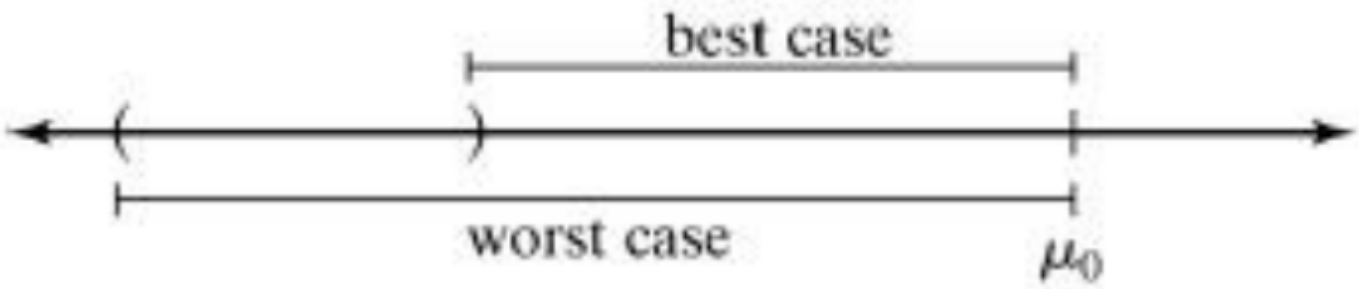
- Confidence interval testing: evaluate whether the simulation and the real system performance measures are close enough.
- If Y is the simulation output, and $\mu = E(Y)$.
- The confidence interval (CI) for μ is:

$$\bar{Y} \pm t_{\alpha/2,n-1} \frac{S}{\sqrt{n}}$$

- Validating the model ϵ is a difference value chosen by the analyst, that is small enough to allow valid decisions to be based on simulations.

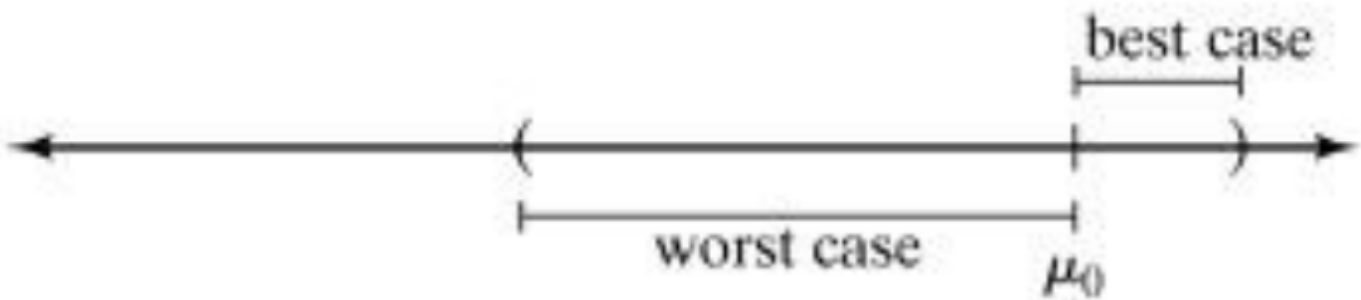
Confidence Interval Testing II

- Suppose the CI does not contain μ_0:
  - If the best-case error is $> \epsilon$, model needs to be refined.
  - If the worst-case error is $\leq \epsilon$, accept the model.
  - If best-case error is ϵ, additional replications are necessary.



Confidence Interval Testing III

- Suppose the CI contains μ_0:
  - If either the best-case or worst-case error is $> \epsilon$, additional replications are necessary.
  - If the worst-case error is ϵ, accept the model.



Confidence Interval Testing - Bank example

- $\mu_0 = 4.3$, and we define that a "close enough difference" is $\epsilon = 1$ minute of expected customer delay.
- A 95 % CI, based on the 6 replications is [1.65, 3.37]:

$$\bar{Y} \pm t_{0.025,5} \frac{S}{\sqrt{n}} = 2.51 \pm 2.5706 \times \frac{0.82}{\sqrt{6}}$$

- $\mu_0 = 4.3$ falls outside the confidence interval
  - the best case $|3.37 - 4.3| = 0.93 < 1$, but
  - the worst case $|1.65 - 4.3| = 2.65 > 1$
  - **Additional replications are needed to reach a decision.**

Summary

- Model validation is essential:
  - Model verification
  - Calibration and validation
  - Conceptual validation
- Need to compare system data to model data, and make comparison using a wide variety of techniques.
- Some techniques that we can use:
  - Guarantee high face validity by consulting knowledgeable persons.
  - Conduct simple statistical tests on assumed distributional forms.
  - Conduct a Turing test.
  - Compare model output to system output by statistical tests.

Sources and Resources

- Kleijnen, J. P. (1995). Verification and validation of simulation models. European Journal of Operational Research, 82(1), 145-162.
- Faculty of Math and CS - UBB Discrete Event Simulations.
- Errors in Hypothesis Testing.