Basics of Simulation and Modelling Methodology

The Art of Simulation

- The most fundamental methodological issue is how to map a real problem into one solvable (to an extent that is possible) by simulation

- Even if possible, would be too costly in terms of complexity or run time

- To reduce the problem to a manageable one within acceptable approximation, the following approaches are useful:

  – **Modelling**, where we are interested in accurately representing the system or specific processes within the system, in the simplest manner possible

  – **Performance Evaluation**, where we are interested in estimating the appropriate performance measure for that system

The Art of Simulation

- *Simulation* is no different from traditional methods of *analysis* in which one approximation or another is almost always made

- However, analysis typically computes a number that represents the quantity of interest but in simulation waveforms unfold in time in what is hoped to be a good imitation of the system of interest simulation can know the process of whole system

- Simulation possesses a dynamic quality absent from analysis, allowing monitoring of the system at different points, providing insight not otherwise available and its flexibility enables to track the evolution

- Simulation is equivalent to a random experiment. That is, the output must be observed for a period of time and statistics collected

Methodology of Prob. Solving for Simulation

- A real communication system generally is far too complex to simulate and the objective is to reduce the complexity of a problem

- Reducing the larger problem into simpler form can be viewed as *conducting a **conditional** experiment*

- Consider the output waveform V_t of a system at discrete time t

$$V_t = g(\Omega)$$

  where g is the system transfer characteristic and $\Omega=(\mathbf{z}_1,\ldots,\mathbf{z}_K)$ is a collection of discrete-time input processes

- The function of a simulation would be to produce a sequence of values $\{V_t\}$ for $t=kT_s$, $k=\pm1, \pm2, \ldots$ with T_s the simulation sampling interval

Methodology of Prob. Solving for Simulation

- A conditional experiment would produce

$$V_t = g(\Omega')$$

  where $\Omega' = (\mathbf{z}_1, \ldots, \mathbf{z}_k, \mathbf{z}_{k+1} = \xi_{k+1}, \ldots, \mathbf{z}_K = \xi_K)$. That is, the first k processes are simulated while the remainder are held at fixed (vector) values

- Conditioning in this sense produces a simpler experiment, or one which is faster, or one whose results are easier to understand

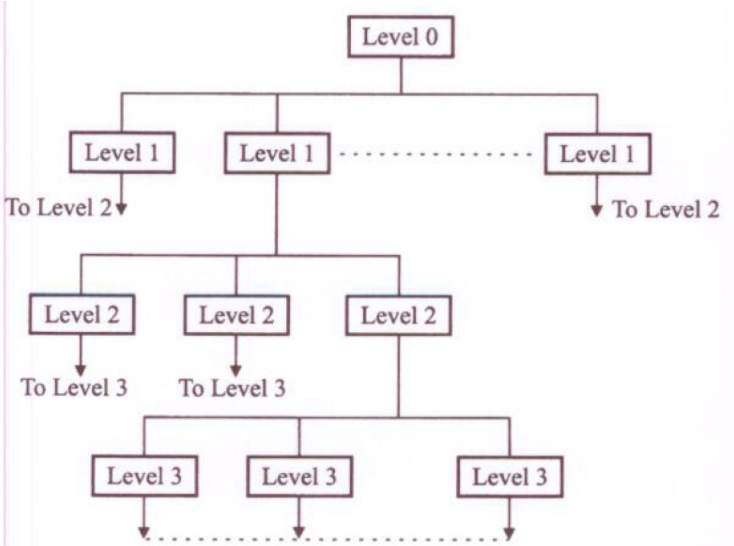
- In general the experiment would have to be repeated for a set of the conditions and unconditioning can be done by a subsequent simulation or by analytical means

Basic Concepts of Modelling

- In analysis, simplified or idealised models are often used for tractability
- In simulation, it is typically no more difficult, although computationally more expensive, to use a more complicated model
- There is a trade-off between accuracy and computer run time
- A system can be viewed as an interconnected set of “subsystems” and referred to as a *block diagram*

Basic Concepts of Modelling

- A complete description of any system can be visualised as a tree diagram with succeeding branches representing ↑ levels of detail
  - A low-level model, i.e., a circuit model, would implement Kirchhoff's equations, representing each component by its differential equation
  - The filter's effect is entirely predictable from the transfer function H(f)



System Modelling

- A system is a communication link which at the highest level of description is represented by a block diagram of subsystems
- At any level of the tree it is also possible to reduce modelling complexity by using only a subset of the blocks at that level
- Some of the subsystems may be completely omitted (we always do!)
  - For instance, synchronisation may be assumed to be perfect
- Another type of subsystem which is typically not simulated is one which converts an analogue signal to a digital signal (A/D converter)

Device Modelling

- A device is simply a block at the subsystem level which contains whatever the system designer wishes
- It could be a piece of manufactured equipment, e.g., simply cabling, waveguide runs or other manufactured media, … etc
- The ideal device model is a transfer function model, i.e., a rule for producing at each instance an output value based on the input values
- How does one arrive at a good rule?
  - This description should accommodate departures from ideal behaviour in ways that are meaningful and physically realisable
  - A good subsystem model should have variable input parameters that can be set to reflect the actual behaviour of devices

Random Process Modelling

- The inputs and outputs of systems and subsystems are desired (information) and undesired (noise and interference) *random processes* and any simulation is to compute

- The imitation is produced by what is called a **random number generator** (RNG) which emits a sequence of numbers that forms a sampled version of a segment of a sample path of the random process

- Information sources and noise sources are both random processes but the test signals may be deterministic, e.g., a sinusoid

- Another type of random process that we may need to model is a “random” channel such as a multipath channel, the CIR h(τ;t) which is typically assumed to be randomly time-varying (our LAB!)

Simulation with Hardware in the Loop

- It is possible to use the actual piece of hardware in the simulation **BUT**

  Disadvantages in hardware simulation

  - The simulation/hardware interface is difficult to realise
  - The simulation samples will have to be fed to a D/A converter and up-converted to the C/F of the device
  - Severe incompatibility between the real-time speed of the simulation and the bandwidth of the device

- This is more likely if the actual device does DSP (or baseband)

  Digital signal processing is more likely used by hardware, since

  - If the transmitted signal is well defined by the standards, it will be easy to simulate its transmission (the models are specified in the standards docs)
  - The receiver algorithms will be executed on a workstation in a simulation mode which is then converted into (C) code, then downloaded to an actual processor on an interface board and a hybrid simulation is carried out using simulated received signals running on the actual processor

Performance Evaluation Techniques (PET)

- Modelling deals with the representation of devices, subsystems and processes for obtaining an estimate of some system-level performance

- For many problems of interest, the run time of simulation can become prohibitively long and modifications of the straightforward Monte-Carlo method will involve:

  how to modify the straightforward Monte-Carlo method:

  – Assumptions/simplifications on the properties of the system
  – Assumptions on the statistical properties of waveforms
  – Clever statistical techniques

Performance Evaluation Techniques

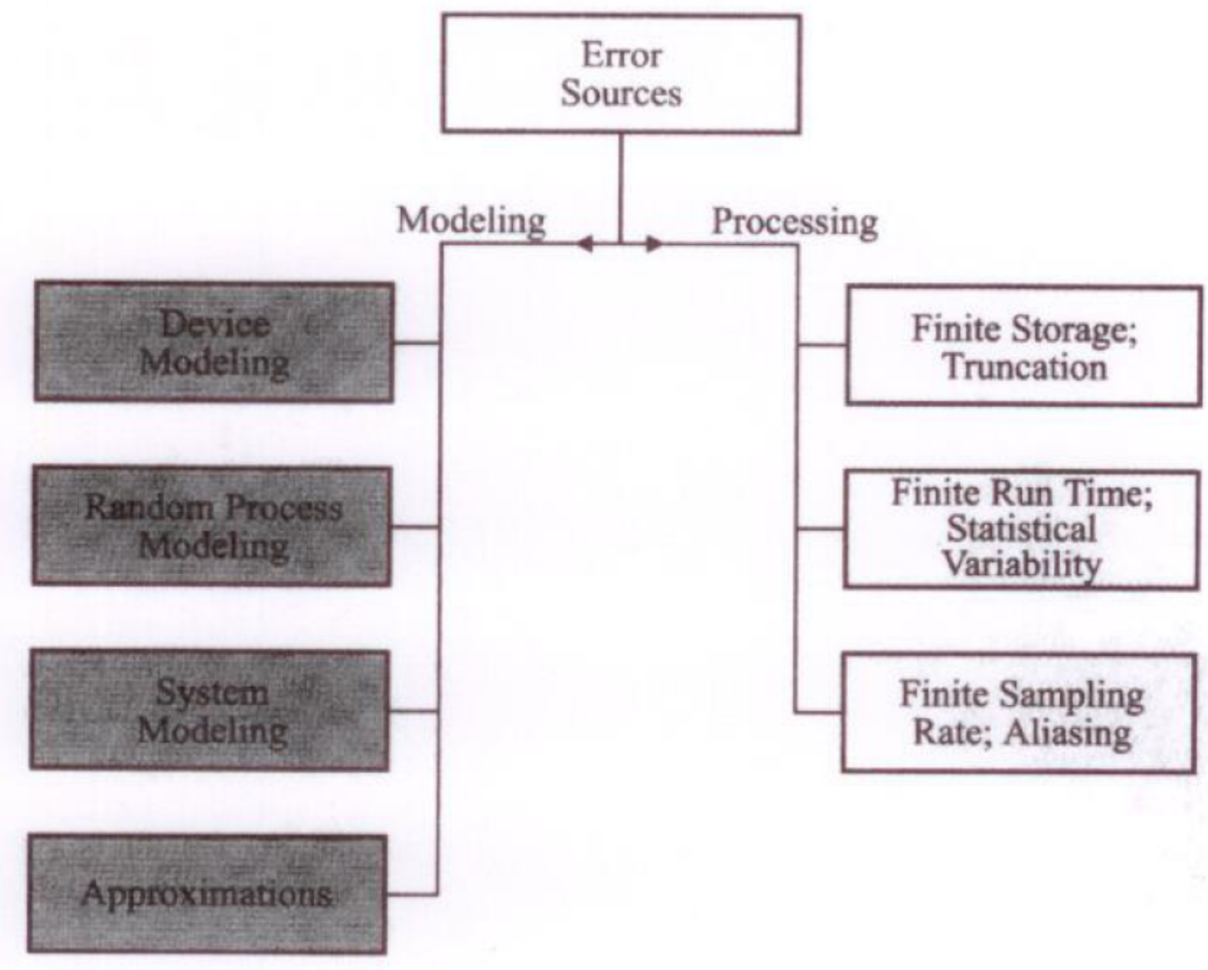
- The measurand of a Monte Carlo simulation (e.g., SNR or BER) is a random variable – the longer we run the simulation the closer it tends to the true value $\rightarrow$ trade-off between run time & accuracy

- The most demanding situation usually is the estimation of BER
  - If a system delivers a BER of 10^{-5} then we expect to see at least one error about every 10^5 bits $\rightarrow$ the rule of thumb is to simulate >10/BER bits

If the BER is 10^-5, we need to simulate 10*10^5 bits in general.

- Common PET techniques are:
  - Quasi-analytical which for instance by combining analytical knowledge and simulation techniques it is possible to simplify the system model
  - A simulation can be carried out by emulating such a sequence of segments each can be visualised as being a conditional experiment

Error Sources in Simulation



Processing Errors

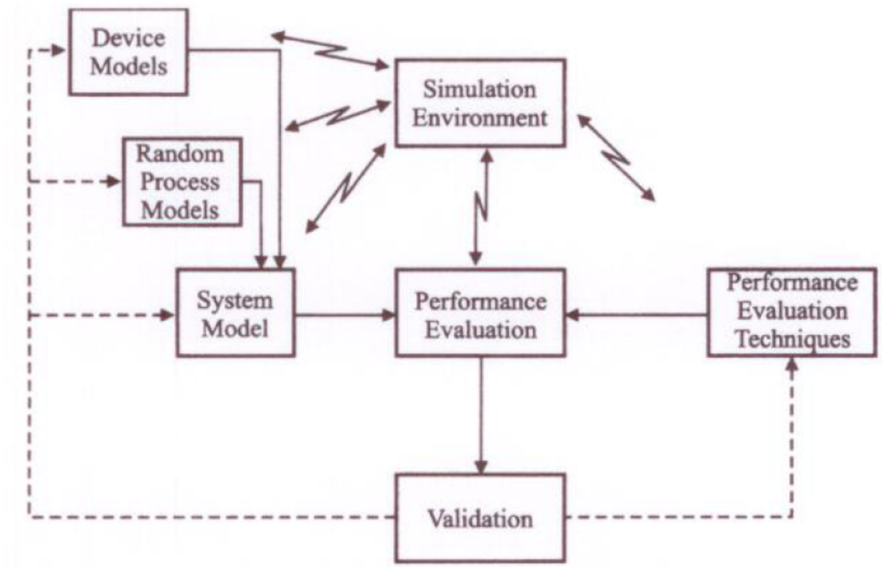
- **"Processing" errors** exist because
  - We have computing limitations
  - Discrete-time representation of continuous signals induces aliasing error
  - Computer memory will not allow storage of infinite impulse responses
  - Numerical representation will not allow unlimited accuracy
  - Run-time limitations will not allow zero statistical uncertainty
- Processing errors are controllable that in principle they can be reduced to arbitrarily small error, subject to some computational cost

Modelling Errors

- We have **modelling errors** in
  - System modelling
  - Device modelling
  - Random process modelling
- The modelling errors are of different natures and a model which is known to be better will often be more complicated

Validation

- The process of certifying that the simulation results are acceptably close to the correct values is known as validation



- The arrows going both into an out of the outer blocks are meant to indicate that the process of validation may be iterative, i.e., if a simulation is declared not to be validated, some aspect of what produced the result must be changed and the process repeated

Important Issues

- ❑ **Sample-by-Sample or Block Processing** – Simulation can be carried out on a sample-by-sample basis or on blocks of samples

- ❑ **Stream-Driven and Event-Driven Simulations** – These techniques are well suited for handling the asynchronous aspects of some of the signal processing operations and protocols in communication systems

- Data-driven simulators do not have any notion of global time built into the simulation but in event-driven simulations, the global simulation time is advanced to the time of the next scheduled event in the event queue before execution of blocks begins

- In packet com systems where the arrival of a new packet triggers processing operations the inter-arrival time can be arbitrary

Important Issues

- ❑ **Time-Domain Versus Frequency-Domain Processing** – Simulation models can be implemented in the time domain or the frequency domain or a mixture of both

- The choice depends on the nature of the system being simulated. In general, nonlinearities and feedback loops are simulated in the time domain using sample-by-sample processing whereas filters can be simulated in the time domain or the frequency domain