1 Modelling and Simulation

1.1 Introduction

This course teaches various aspects of computer-aided modelling for the performance evaluation of computer systems and communication networks. The performance of a system influences its design, procurement and subsequent use. Performance analysis may be carried out at any stage of that life-cycle. In general, the problem is to achieve the highest possible performance given the constraints on the system. Typical constraints might be the number of users, the bandwidth or the cost. Modelling and simulation are methods which are commonly used by performance analysts to represent constraints and optimise performance.

Many of the ideas of the course will be presented by examples; as well as computer systems and communication networks, other dynamic systems such as manufacturing and transport systems will sometimes be considered. With all the examples it will be as important to consider the underlying concepts as the details of the example itself.

1.2 Models as tools

The term *model* conjures up a variety of images. We will use it to mean something which represents a system in a form which allows us to make predictions about the behaviour of the system. How accurate these predictions are will depend on the detail we invest in the model. In some cases, when we only need an approximate or relative measure of the behaviour of the system, a very crude model will suffice. When the answers obtained from the model are crucial we will be prepared to expend more effort in developing our model and the representation of the system will be less abstract. During the course we will consider a range of modelling techniques from the very abstract, in which the behaviour of the system is captured by a single equation, the so-called *operational laws*, to the very detailed *simulation models* in which every aspect of the system's behaviour can be represented.

In all cases a model should be an abstraction of the system: an attempt to distil, from the mass of details that is the system itself, exactly those aspects that are essential to the system's behaviour. Once a model has been *defined* through this abstraction process, it can be *parameterised* to reflect any of the alternatives under study, and then *evaluated* to determine its behaviour under this alternative. Using a model to investigate system behaviour is less laborious and more flexible than direct experimentation, because the model is an abstraction that avoids unnecessary detail.

There are many reasons why investigations into the behaviour of a system cannot be carried out by direct experimentation on the system. At best experimentation will probably be disruptive, at worst, dangerous. For example, a system manager of a heavily loaded file server is unlikely to allow experimentation, or even simple monitoring, because the disruption to users will be too great. Systems such as nuclear reactor control systems or the London Underground control system are simply not safe to experiment with. However, timely response may be just as critical as correctness within such systems.

Another problem with direct experimentation may be that the internal behaviour which we wish to investigate may not be accessible in a working system. For example, in most

operating systems it is difficult to obtain the exact timing of instruction level events. Experimentation, since it involves monitoring, is expensive (in terms of system resources) and time consuming. It is also difficult to use the results of direct experimentation to extrapolate to other scenarios of operation. For example, experimenting with one configuration of a communication network is unlikely to give us much insight into the expected performance if another configuration is used. This leads us to one of the most compelling reasons for using modelling instead of direct experimentation: it is often the case that the system which we wish to study does not yet exist.

The idea that a design should be subjected to some analysis to ensure that the resulting system will behave *correctly* is now widely accepted. Models are tools commonly used for this purpose: for example a process algebra, such as CCS, may be used to check that an implementation will satisfy conditions set out in a behavioural specification. Such models are sometimes called *qualitative models* because all definite values are abstracted away. In contrast, performance evaluation is concerned with ensuring that a system will satisfy constraints on timely behaviour and efficient resource utilisation. These constraints are sometimes expressed as *performance requirements* or a *performance specification*. These performance models are sometimes called *quantitative models*—if we wish to extract quantitative information (estimates of real values such as response time or throughput) then we must incorporate quantitative information in the model.

Throughout the course we will encounter a variety of modelling styles. The key notions of the course will recur whichever modelling style we are using. These notions are that

- a model can be constructed to represent some aspect of the dynamic behaviour of a system; and
- once constructed, such a model becomes a tool with which we can investigate the behaviour of the system.

1.3 Discrete event systems

A common feature of all the systems which we will consider is that they are *discrete event* systems. What this means is that our view of the system (its *state*) is characterised by variables that take distinct values and which change at distinct times or *events* in the system. For example, we might be interested in the number of nodes in a wireless network which are currently waiting to send a message (N say). This number will change when one of two events occur: either a node, which was not previously waiting, generates a message and is now waiting to send ($N \to N + 1$), or a node, which was previously waiting, successfully transmits its message ($N \to N - 1$).

We will not consider *continuous* systems. These are systems in which the state variables characterising the system are constantly changing. An example of such a model would be the speed of a falling object or the drag experienced by a plane.

In all our modelling studies one of our major tasks will be to choose the parameters which *characterise* the system under study.

Discrete time discrete event models only consider the system at *predetermined moments in time*, which are typically evenly spaced, eg. at each clock "tick". State changes are observed only at these moments, which are not necessarily the moment at which the events

themselves occur. Such models are often used to approximate systems with continuously changing states. *Continuous time* discrete event models, consider the system at the *time of each event*. Hence the time parameter in such models is conceptually continuous. At levels of abstraction above the hardware clock these models are generally appropriate for computer and communication systems.

1.4 Performance Evaluation

Performance evaluation is concerned with the description, analysis and optimisation of the dynamic behaviour of computer and communication systems. This involves the investigation of the flow of data, and control information, within and between components of a system. The aim is to understand the behaviour of the system and identify the aspects of the system which are sensitive from a performance point of view. More specifically, a performance analyst might be involved in any of the following tasks: specifying performance requirements, evaluating design alternatives, comparing two or more systems, determining the optimal value of a parameter (*system tuning*), finding the performance bottleneck (*bottleneck identification*), characterising the load on the system (*workload characterisation*), determining the number and sizes of components (*capacity planning*), and predicting the performance at future loads (*forecasting*).

These tasks address problems which are regularly faced by organisations during the life-time of a computer or communication system. For example, consider the following scenarios:

Design and Implementation A car manufacturing company is designing and building a computer-aided design system to allow thirty car designers simultaneous access to a distributed database through GPU-enhanced PCs. Early in the design phase, fundamental decisions must be made on issues such as the database accessing mechanism and the process synchronisation and communication mechanism. The relative merits of various mechanisms must be evaluated prior to implementation.

Evolution of the configuration and workload A stock exchange intends to begin trading a new class of financial instruments. When this occurs, the exchange's total volume of transactions is expected to increase by a factor of three. Adequate resources, both computer and personnel, must be in place when the change is implemented.

We can use a variety of *measures*, or *indices*, to characterise the behaviour of the system. Perhaps the most commonly used are *response time*, *throughput* and *utilisation*. Each of these can be regarded as a random variable of the model. The response time is an estimate of the time which elapses between the arrival of a request to the system and the completion of the request by the system. Notice that this is quite a vague definition. This is because response time is a vague notion which should be defined more precisely in terms of the system and the goals of the study whenever it is used. Throughput is the number of requests satisfied per unit time. In both these definitions "request" is used as a general term which, again, should be made precise in the context of any particular model. Utilisation is the percentage of time that a system or component spends in use, as opposed to being idle.

Other measures of interest might be the total number of requests over a given time period, the mean number of requests waiting, or the mean time a request spends waiting while the system is otherwise occupied, the proportion of requests that are satisfied and the proportion that are discarded (blocked or lost).

1.5 Probability Theory

Although computer systems by their very nature are deterministic systems the models we will be considering will be *stochastic*. Loosely speaking, a stochastic model is one in which behaviour is random. We use such models because the world that computers interact with is by no means deterministic, but instead full of random behaviour. Also, the stochastic nature of the models will allow us to consider typical or average behaviour rather than particular behaviour. In other words, we use probability distributions as a form of abstraction. We will rely heavily on basic probability theory when constructing and solving our models.

Probability theory deals with the analysis of events which are unpredictable when viewed individually, but which exhibit some regularity and predictability when viewed in large numbers. We are familiar with such ideas applied to tossing a coin or rolling a die. Here the unpredictability of an individual outcome is due to the non-deterministic nature of the situation. When we are considering a computer system the unpredictability of an individual outcome is likely to arise because of our inability to measure or specify the exact conditions under which the event occurs. This distinction, however, is largely philosophical; probability theory is equally applicable to both these situations.

There are three axioms of probability theory:

1. There is a *sample space*, Ω, which encompasses all possible observations or outcomes.
2. There is a collection of subsets of Ω, denoted E, and termed *events*[1]; these subsets are usually identified as sample points which satisfy some condition.
3. There is a *probability mapping*, Pr, from E to $\mathbb{R}$. Pr must satisfy three simple conditions:
   (a) For any event A, $A \in E$, the mapping Pr is defined and satisfies $0 \leq \Pr(A) \leq 1$.
   (b) $\Pr(\Omega) = 1$.
   (c) If A and B are mutually exclusive, that is, they contain no sample points in common, then $\Pr(A \cup B) = \Pr(A) + \Pr(B)$.

If we consider a mobile phone antenna using frequency division multiplexing the sample space is the complete set of observations. Example events might be that more than three frequencies are in use or that a particular user is making use of the facility. These events corresponds to subsets of the sample points.

Various properties of probabilities can be derived from the axioms and simple set theory. For example, the probability of the union of two events A and B which are **not** mutually exclusive is $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$. Similarly, the probability of the complement of event A, denoted by $\neg A$, is $\Pr(\neg A) = 1 - \Pr(A)$.

[1]Note that this is a different use of the term "event" from discrete *event* system.

1.5.1 Conditional Probabilities

If we know that some event has occurred then this may affect the probability that other events have occurred. The *conditional probability* of A occurring, given that B has occurred, is

$$\boxed{\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)}}$$

Example: In a communication network, there are two routes A and B, between a particular pair of nodes which are chosen with equal probability. Message transmission time along route A is 1 second and along route B is 3 seconds. However, the probability of a transmission error is 0.2 on route A and 0.1 on route B. We wish to find the probability of a message being transmitted correctly in 1 second.

Let us denote the event of a successful message transmission by C. We wish to calculate the probability that a transmission takes route A and is successful, i.e. $\Pr(A \cap C)$. We know the conditional probability that a transmission is successful, given that it took route A, is $\Pr(C \mid A) = 0.8$ and the probability that route A is taken is $\Pr(A) = 0.5$. So it follows that $\Pr(A \cap C) = \Pr(C \mid A) \times \Pr(A) = 0.8 \times 0.5 = 0.4$.

If A and B are mutually exclusive $\Pr(A \mid B) = 0$. If B is a precondition for A, then $\Pr(A \cap B) = \Pr(A)$. Two events are *independent* if knowledge of the occurrence of one of them tells us nothing about the probability of the other, i.e. $\Pr(A \mid B) = \Pr(A)$. Thus, when two events are independent, the probability of them both occurring is

$$\boxed{\Pr(A \cap B) = \Pr(A) \times \Pr(B).}$$

1.5.2 Random Variables and Probability Distributions

Often when we conduct an experiment or investigation we are interested in some number associated with the experiment rather than the actual outcome. For example, if we consider the mobile phone antenna again, we are likely to be interested in the number of frequencies in use rather than the details of each connection. Thus we naturally construct a mapping from the observations we make of the actual interactions, the *sample space*, to the value we are interested in. Such a function, from the sample space to the real or natural numbers is a continuous or discrete *random variable*, respectively. We are often interested in calculating, for each possible value, with what probability the random variable takes that value. Conversely, knowing these probabilities gives us a lot of information about the sample space modelled by the random variable.

When the random variable takes discrete values such as the natural numbers we associate probabilities with the random variable in an intuitive way. That is, the probability that a random variable X has the value v is equal to the probability of the union of all events that lead to $X = v$.

Suppose that random variable X has values $\{x_i \mid i \in I\}$ and that for all $i \in I$ the probability that $X = x_i$ is equal to p_i. Then $\{p_i \mid i \in I\}$ is the *probability distribution* of X. This mapping from x_i to p_i is sometimes called the *mass function*. In the simple example

of rolling a fair die, the values of the random variable are clearly $\{1, 2, 3, 4, 5, 6\}$ and the probability distribution is $\{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}$. This is a *uniform* distribution.

The *expectation* of X is defined to be

$$\boxed{E[X] = \sum_{i \in I} x_i \times p_i.}$$

Informally, we can think of this as the average value that X will have. For rolling a fair die this will be

$$(1 \times 1/6) + (2 \times 1/6) + (3 \times 1/6) + (4 \times 1/6) + (5 \times 1/6) + (6 \times 1/6) = 21/6 = 3.5$$

For some sample spaces, discrete values of random variables are not appropriate. For example, if we are interested in the number of frequencies which are used each hour then a discrete random variable is ideal. However, if we are interested in the duration of each connection, the variable can take almost any non-negative value, so a continuous random variable is appropriate.

For continuous random variables we can no longer assign a probability to each sample point in a meaningful way. Instead we define the probability that the random variable takes a value which is less than a given value. This is done by a *distribution function*[2], F. For a continuous random variable X the distribution function is defined as

$$\boxed{F(x) = \Pr(X \leq x)}$$

Probability distribution functions satisfy four conditions:

1. $F(x_1) \leq F(x_2)$ if $x_1 \leq x_2$;
2. $F(x) \longrightarrow 0$ as $x \longrightarrow -\infty$;
3. $F(x) \longrightarrow 1$ as $x \longrightarrow \infty$;
4. $F(x)$ is continuous[3].

The derivative of the distribution is known as the *density function* $f(x)$,

$$\boxed{f(x) = \frac{dF(x)}{dx}.}$$

It is convenient to regard the infinitesimal quantity $f(x)dx$ as the probability that the value of X lies in the interval $(x, x + dx)$. The *expectation* of the continuous random variable X is defined as

$$\boxed{E[X] = \int_{-\infty}^{\infty} x f(x) dx.}$$

[2]Some authors call this the *cumulative distribution function*.

[3]Strictly, $F(x)$ only needs to be *right continuous*, but we will assume that $F(x)$ is continuous for all the continuous random variables which we will encounter in this course.

The expectation of the sum of a number of random variables is the sum of their individual expectations. This is true even if the variables are dependent in some way. In other words, for continuous random variables, X and Y, $E[X+Y] = E[X] + E[Y]$.

The expectation only gives us an idea of the average value assumed by a random variable, not how much individual values may differ from this average. The *variance*, $Var(X)$, gives us an indication of the "spread" of values:

$$\boxed{Var(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2.}$$

1.5.3 Two important distributions

The most important continuous probability distribution for performance modelling purposes is the *(negative) exponential distribution* which has the distribution function

$$F(x) = 1 - e^{-\mu x} \quad x \geq 0 \qquad \text{and density function} \qquad f(x) = \mu e^{-\mu x} \quad x \geq 0$$

The expectation of an exponential random variable with parameter μ is $1/\mu$ and its variance is $1/\mu^2$. This random variable is often used to denote the time which elapses until some event occurs, such as the arrival of a task at a computer system, the arrival of a bus at a bus-stop or the outbreak of war.

Example: Suppose that the time for which a computer system is up is an exponential random variable with mean 20 days. What is the probability that the machine will stay up for more than 40 days?

Let X denote the random variable representing the up-time of the machine. Then we are interested in $\Pr(X > 40) = 1 - \Pr(X \leq 40)$ (i.e. the distribution function evaluated when $x = 40$ and $\mu = 1/20$):

$$\Pr(X > 40) = 1 - (1 - e^{-40/20}) = e^{-2} = 0.135 \qquad \text{to 3 significant figures}$$

The exponential distribution function is closely related to a discrete random variable, the *Poisson* distribution. This random variable takes values in the set $\{0, 1, 2, \ldots\}$ and has the mass function

$$p_i = e^{-\mu}\frac{\mu^i}{i!} \qquad i \geq 0.$$

The expectation of a Poisson random variable with parameter μ is also μ. The Poisson random variable arises frequently in computer and communication system modelling. It is typically used as a counting variable, recording the number of events that occur in a given period of time; for example, the number of sms messages transmitted from a mobile phone in one hour or the number of database queries arriving at a server in one minute. If we observe a Poisson process with parameter μ for some time period of length h then:

- the probability that one event occurs is $\mu h + o(h)$.[4]

[4]The notation $o(h)$ denotes terms of small relative size; formally a function f is $o(h)$ if $\lim_{h\to 0} \frac{f(h)}{h} = 0$ In practice, what this means is that these are terms which we can ignore when h is small.

- the probability that two or more events occur is $o(h)$.
- the probability that no events occur is $1 - \mu h + o(h)$.

Thus if we observe a Poisson process for a infinitesimal time period dt the probability that an event occurs is μdt. If the occurrence of events is governed by a Poisson distribution then the inter-event times are governed by an exponential distribution with the same parameter, and vice versa. Therefore, if we know that the delay until an event is exponentially distributed then the probability that it will occur in the infinitesimal time interval of length dt, is μdt.

The reason that the exponential distribution is used extensively in performance modelling is because it has several attractive mathematical features. Here we will state those properties without proof, but the proofs can be found in any textbook on stochastic modelling, Markov processes or performance modelling.

- The exponential distribution is *memoryless*: if X is an exponentially distributed random variable, then

  $$\Pr(X > t + s | X > t) = \Pr(X > s) \qquad t > 0, s > 0$$

  If X represents the waiting time until an event, this implies that knowing that t time units have already elapsed without the event occuring does not give us any additional information about when the event will occur—the distribution of the further waiting time is the same as if no waiting time had already passed.
- If X_1 and X_2 are two exponentially distributed random variables, with parameters λ_1 and λ_2 respectively, and Y is defined to be their minimum, i.e. $Y = \min(X_1, X_2)$, then Y is also exponentially distributed, with parameter $\lambda_1 + \lambda_2$. This is sometimes called the *superposition* property.
- Let X be an exponentially distributed random variable, with parameter λ, representing time until an event. Suppose that the events are probabilistically divided into two categories, with events belonging to stream A with probability p_A and belonging to stream B with probability $p_B = 1 - p_A$. Then stream A and stream B are Poisson streams with parameters $\lambda \times p_A$ and $\lambda \times p_B$ respectively, and the waiting times between events of category A and between events of category B are exponentially distributed with parameters $\lambda \times p_A$ and $\lambda \times p_B$ respectively. This is sometimes called the *decomposition property.*

1.6 The rest of the course

The central theme of the course will be that a *model*, as well as being an abstract representation of a system, is a tool which we can exploit to derive information about the system. The more detail we invest in the model, the more sophisticated the information we can extract from it. As the course progresses the models will become increasingly detailed; the corresponding solution techniques will similarly become more complex, relying on increasing levels of computer assistance.

Jane Hillston ⟨Jane.Hillston@ed.ac.uk⟩. January 5, 2017.