1

Introduction, Scope, Definitions

Preview

This chapter attempts to motivate the student for the course. Why should s/he study modeling and simulation? What can these techniques do for him or her that other techniques might not? We shall start out with some basic definitions of terms that will be used in this text over and over again, such as the terms "system", "experiment", "model", and "simulation". We shall then discuss good and bad reasons for using modeling and simulation as problem solving tools. We shall finally list areas of science and engineering to which modeling and simulation have been successfully applied, and we shall explain what makes these various application areas different from each other, and why the modeling and simulation approaches taken in these application areas vary so drastically from each other.

1.1 What is a System?

What is it that we focus on when we talk about a "system"? Brian Gaines gave the following interesting (and verbose) definition of what a "system" is [1.2]:

> " 'A system is what is distinguished as a system.' At first sight this looks to be a nonstatement. Systems are whatever we like to distinguish as systems. Has anything been said? Is there any possible foundation here for a systems science? I want to answer both these questions affirmatively and show that this definition is full of content and rich in its interpretation.

"Let me first answer one obvious objection to the definition above and turn it to my advantage. You may ask, 'What is peculiarly systemic about this definition?' 'Could I not equally well apply it to all other objects I might wish to define?' i.e.,

"A rabbit is what is distinguished as a rabbit. 'Ah, but,' I shall reply, 'my definition is adequate to define a system but yours is not adequate to define a rabbit.' In this lies the essence of systems theory: that to distinguish some entity as being a system is a necessary and sufficient criterion for its being a system, and this is uniquely true for systems. Whereas to distinguish some entity as being anything else is a necessary criterion to its being that something but not a sufficient one.

"More poetically, we may say that the concept of a system stands at the supremum of the hierarchy of being. That sounds like a very important place to be. Perhaps it is. But when we realize that getting there is achieved through the rather negative virtue of not having any further distinguishing characteristics, then it is not so impressive a qualification. I believe this definition of a system as being that which uniquely is defined by making a distinction explains many of the virtues, and the vices, of systems theory. The power of the concept is its sheer generality; and we emphasize this naked lack of qualification in the term general systems theory, rather than attempt to obfuscate the matter by giving it some respectable covering term such as mathematical systems theory. The weakness, and paradoxically the prime strength of the concept is in its failure to require further distinctions. It is a weakness when we fail to recognize the significance of those further distinctions to the subject matter in hand. It is a strength when those further distinctions are themselves unnecessary to the argument and only serve to obscure a general truth through a covering of specialist jargon. No wonder, general systems theory is subject to extremes of villification and praise."

Brian Gaines expresses here in very nice words simply the following: The largest possible system of all is the universe. Whenever we decide to cut out a piece of the universe such that we can clearly say what is *inside* that piece (belongs to that piece), and what is *outside* that piece (does not belong to that piece), we define a new "system".

A system is characterized by the fact that we can say what belongs to it and what does not, and by the fact that we can specify how it interacts with its environment. System definitions can furthermore be hierarchical. We can take the piece from before, cut out a yet smaller part of it, and we have a new "system".

Let me quote another famous definition which is due to Ross Ashby [1.1]:

> "At this point, we must be clear about how a 'system' is to be defined. Our first impulse is to point at the pendulum and to say 'the system is that thing there'. This method, however, has a fundamental disadvantage: every material object contains no less than an infinity of variables, and therefore, of possible systems. The real pendulum, for instance, has not only length and position; it has also mass, temperature, electric conductivity, crystalline structure, chemical impurities, some radio–activity, velocity, reflecting power, tensile strength, a surface film of moisture, bacterial contamination, an optical absorption, elasticity, shape, specific gravity, and so on and on. Any suggestion that we should study 'all' the facts is unrealistic, and actually the attempt is never made. What is necessary is that we should pick out and study the facts that are relevant to some main interest that is already given ...
>
> "... The system now means, not a thing, but a list of variables."

Clearly, the two definitions are in contradiction with each other. According to the former definition (by Brian Gaines), the pendulum certainly qualifies for a system, and I would agree with him on that. However, taking the pendulum, we can now "cut out" a smaller piece by declaring that we are only interested in certain properties of the pendulum, say: its mass and its length, and thereby define another "system". The "cutting" does not necessarily denote a separation in the physical world, it can also take place at the level of a mathematical abstraction ... and in the context of modeling, this is actually most commonly the case.

Another property of a "system" is the fact that it can be "controlled" and "observed". Its interactions with the environment naturally fall into two categories:

(1) There are variables that are generated by the environment, and that influence the behavior of the system. These are called the *"inputs"* of the system.

(2) There are other variables that are determined by the system, and that in turn influence the behavior of its environment. These are called the *"outputs"* of the system.

In general, we ought to be able to assign values to at least some of the "inputs" of the system, and observe the behavior of the system by recording the resulting "outputs".

This leads to yet another definition for the term "system":

> "A system is a potential source of data."

I personally like this definition best, because it is very short and concise. I am not entirely sure who gave this definition first, but I believe it was Bernard Zeigler [1.9].

1.2 What is an Experiment?

The last definition for "system" immediately leads to a definition for the term "experiment":

> "An experiment is the process of extracting data from a system by exerting it through its inputs."

Experimenting with a system thus means to make use of its property of being "controllable" and "observable" (please, notice that these terms are used here in a plausible sense rather than in the more stringent sense of linear system theory [1.3]). To perform an experiment on the system means to apply a set of external conditions to the accessible inputs, and to observe the reaction of the system to these inputs by recording the trajectory behavior of the accessible outputs.

One of the major disadvantages of experimenting with real systems is the fact that these systems usually are under the influence of a large number of additional inaccessible "inputs" (so–called "disturbances"), and that a number of really useful "outputs" are not accessible through measurements either (they are internal "states" of the system).

One of the major motivations for "simulation" is the fact, that, in the simulation world, *all* "inputs" and "outputs" are accessible. This allows us to execute simulation runs that lie outside the range of experiments that are applicable to the real system.

1.3 What is a Model?

Given the above definitions for "systems" and "experiments", we can now attempt to define what we mean by the term "model". I shall give the definition that was first coined by Marvin Minsky [1.7]:

> "A model (M) for a system (S) and an experiment (E) is anything to which E can be applied in order to answer questions about S."

Notice that this definition does not imply that a "model" is a computer program. It could as well be a piece of hardware or simply an understanding of how a particular system works (a so-called "mental model"). However, in this text, we shall concentrate on the subclass of models that are codable as computer programs (the so-called "mathematical models").

Notice that the above definition clearly qualifies any "model" to be called a "system". This automatically implies that models are hierarchical in nature, i.e., we can "cut" a smaller portion out, and thereby generate a new model which is valid for a subset of the experiments for which the original model was valid. It is thus common to create models of models. Jack Kleijnen calls such models "meta-models" [1.5]. Bernard Zeigler talks about "pruning" particular features out of a model to create a simplified version of the previous model [1.11].

Notice finally that the above definition does not describe "models for systems" *per se.* A model is always related to the tuple *system and experiment.* If someone says that "a model of a system is invalid" (as can be frequently read), s/he does not know what s/he is talking about. A model of a system may be valid for one experiment, and invalid for another, that is: the term "model validation" always relates to an experiment or class of experiments to be performed on a system, rather than to the system alone. Clearly, *any* model is valid for the "null experiment" applied to *any* system (if we don't want to get *any* answers out of a given simulation, we can use any model for that purpose). On the other hand, no model of a system is valid for *all* possible experiments except the system itself or an identical copy thereof.

1.4 What is a Simulation?

Again, many definitions exist for the term "simulation", but I shall quote the one that I like best. It has been coined by Granino Korn [1.6]:

"A simulation is an experiment performed on a model."

As before, this definition does not imply that the simulation is coded in a computer program. However, in this text, we shall concentrate on the subset of simulations which are codable as computer programs (the so–called "mathematical simulations").

A *mathematical simulation* is a coded description of an experiment with a reference (a pointer) to the model to which this experiment is to be applied.

It was Bernard Zeigler who first pointed out the importance of the physical separation between the *model description* on the one hand, and the *experiment description* on the other [1.9]. We want to be able to experiment with models as easily and conveniently as with real systems. We want to be able to use our simulation tool in exactly the same way as we would use an oscilloscope in the lab.

However, a certain danger lies in this separation. It makes it all too easy to apply an experiment to a model for which the model is not valid. In the lab environment, this can never happen since the real system is valid for *all* experiments, whereas the model is not. Bernard Zeigler realized this problem, and therefore demanded that the model description contain, as an intrinsic and unseparable part, an *experimental frame* definition [1.9]. The "experimental frame" establishes the set of experiments for which the model is valid. When a simulation refers to that model, the actual experiment is then compared with the experimental frame of the model, and the execution of the simulation will only be allowed if the simulation experiment to be performed is established as belonging to the set of applicable experiments.

Unfortunately, today's realities don't reflect these conceptual demands very well. Most commercially available simulation software systems are *monolithic*. They do not support the concept of separating the model description from the experiment description. While the model description mechanisms have seen quite a bit of progress over the years, most software systems do not allow the user to specify models in a truly hierarchical manner. Mechanisms for describing simulation experiments are meager, and mechanisms for describing experimental frames barely exist. In fact, we are still lacking an appropriate "language" to express experimental frames in general terms. All this still belongs to the area of open research.

1.5 Why is Modeling Important?

Let me quote yet another definition of the term "modeling" which is also attributed to Bernard Zeigler [1.10].

> "Modeling means the process of organizing knowledge about a given system."

By performing experiments, we gather knowledge about a system. However, in the beginning, this knowledge is completely unstructured. By understanding what are *causes* and what are *effects*, by placing observations both in a *temporal* as well as a *spatial* order, we organize the knowledge that we gathered during the experiment. According to the above (very general) definition, we are thereby engaged in a process of *modeling*. No wonder that every single discipline of science and engineering is interested in modeling, and utilizes modeling as a problem solving tool.

It can thus be said that modeling is the single most central activity that unites *all* scientific and engineering endeavor. While the scientist is happy to simply *observe* and *understand* the world, i.e., create a model of the world, the engineer wants to *modify* it to his or her advantage. While science is all *analysis*, the essence of engineering is *design*. As this text will demonstrate, *simulation* can be used not only for analysis (the so-called *direct problems*), but also for design (the so-called *inverse problems*).

1.6 Why is Simulation Important?

Except by experimentation with the real system, *simulation* is the only technique available for the analysis of arbitrary system behavior. *Analytical techniques* are great, but they usually require a set of simplifying assumptions to be made before they become applicable; assumptions that cannot always be justified, and even if they might be, whose justification cannot be verified except by experimentation or simulation. In other words, simulation is often not used alone, but in an interplay with other analytical or semi–analytical techniques.

The typical scenario of a scientific discovery is as follows:

(1) The scientist performs experiments on the real system to extract data (to gather knowledge).

(2) S/he then looks at the data, and postulates a number of hypotheses relating to the data.

(3) S/he makes simplifying assumptions to make the data tractable by analytical techniques to test these hypotheses.

(4) S/he then performs a number of simulation runs with different experimental parameters to verify that the simplifying assumptions were justified.

(5) S/he performs the analysis of her or his system, verifies (or modifies) the hypotheses, and finally draws some conclusions.

(6) S/he again performs a number of simulation runs to verify the conclusions.

Most of today's simulation software systems live in isolation. They do not allow us to easily combine simulation studies with other techniques applied to the same set of data. Even the data gathered in the experiment must often be retyped (or at least edited) to fit into the framework of the simulation software system. This text, however, presents the reader with tools (CTRL–C and MATLAB) that are much more flexible than the old–fashioned CSSL–type simulation languages (such as ACSL or DARE–P), and that strongly support a mixed environment of different analysis techniques *including* simulation as just one of them.

Simulation is applicable where other analytical techniques are not. Since such situations are very common, simulation is often the only game in town. No wonder that simulation is the single most frequently used problem solving tool throughout all disciplines of science and engineering.

1.7 The Dangers of Simulation?

The most important strengths of simulation, but also ironically its most serious drawbacks, are the generality and ease of its applicability. It does not require much of a genius to be able to utilize a simulation program. Every stupe can do that ... but that does not make him or her less stupid. In order to use simulation *intelligently*, we must understand what we are doing.

All too often, simulation is a love story with an unhappy ending. We create a model of a system, and then fall in love with it. Since love is usually blind, we immediately forget all about the experimental frame, we forget that this is *not* the real world, but represents the world only under a very limited set of experimental conditions (we become "model addicts"). We find a control strategy that "shapes" our model "world" the way we want it to be, and then apply that control strategy back to the real world, convinced that we now have the handle to make the real world behave the way we want it to ... and here comes the unhappy ending — it probably doesn't.

In this way, the Australians introduced the rabbit to the continent ... and found that these rabbits were soon all over the place and ate all the food that was necessary for the local species to survive ... since the rabbits did not have a natural enemy. Thereafter, the Australians introduced the fox to the continent (foxes feed on rabbits — the "model") ... to see that the foxes were soon all over the place ... but left the rabbits alone ... since they found the local marsupialia much easier to hunt.

Simulations are rarely enlightening. In fact, running simulations is very similar to performing experiments in the lab. We usually need *many* experiments, before we can draw legitimate conclusions. Correspondingly, we need *many* simulations before we understand how our model behaves. While other analytical techniques (where they are applicable) often provide an understanding as to how a model behaves under *arbitrary* experimental conditions, one simulation run tells us only how the model behaves under the one set of experimental conditions applied during the simulation run.

Therefore, while other analytical techniques are generally more restricted (they have a much smaller domain of applicability), they are more powerful where they apply. So, whenever we *have* a valid alternative to simulation, we should, by all means, make use of it. Only a stupe uses simulation *in place* of other analytical techniques.

1.8 Good Reasons to Use Simulation

Let me state a number of good reasons for using simulation as a problem solving tool.

(1) The physical system is not available. Often, simulations are used to determine whether a projected system should ever be built. So obviously, experimentation is out of the question. This is common practice for *engineering* systems (for example: an electrical circuit) with well established and widely applicable meta–knowledge. It is very dangerous to rely on such a decision in the case of systems from soft sciences (the so–called *ill–defined systems*) since the meta–knowledge available for these types of systems is usually not validated for an extension into unknown territory.

(2) The experiment may be dangerous. Often, simulations are performed in order to find out whether the real experiment might "blow up", placing the experimenter and/or the equipment under danger of injury/damage or death/destruction (for example: an atomic reactor, or an aircraft flown by an inexperienced person for training purposes).

(3) The cost of experimentation is too high. Often, simulations are used where real experiments are too expensive. The necessary measurement tools may not be available, or are expensive to buy. It is possible that the system is used all the time, and taking it "off–line" would involve unacceptable cost (for example: a power plant, or a commercial airliner).

(4) The time constants (eigenvalues) of the system are not compatible with those of the experimenter. Often, simulations are performed because the real experiment executes so quickly that it can hardly be observed (for example: an explosion), or because the real experiment executes so slowly that the experimenter is long dead before the experiment is completed (for example: a transgression of two galaxies). Simulations allow us to speed up or slow down experiments at will.

(5) Control variables (disturbances), state variables, and/or system parameters may be inaccessible. Often, simulations are performed because they allow us to access *all* inputs and *all* state variables, whereas, in the real system, some inputs (disturbances) may not be accessible to manipulation (for example: the time of sunrise), and some state variables may not be accessible

to measurement. Simulation allows us to manipulate the model outside the feasible range of the physical system. For example, we can decide to change the mass of a body at will from 50 *kg* to 400 *kg*, and repeat the simulation at the stroke of a key. In the physical system, such a modification is either not feasible at all, or it involves a costly and lengthy alteration to the system.

(6) Suppression of disturbances. Often, simulations are performed because they allow us to suppress disturbances which are unavoidable in the real system. This allows us to isolate particular effects, and may lead to a better insight (intuition) into the generic system behavior than would be possible through obscured measurements taken from the real process.

(7) Suppression of second order effects. Often, simulations are performed because they allow us to suppress second order effects (such as non–linearities of system components). Again, this can help with the understanding of the primary underlying functionality of the system.

1.9 The Types of Mathematical Models

What types of mathematical models do exist? A first category is the set of *continuous–time models*. Fig.1.1 shows how a state variable x changes over time in a continuous–time model.

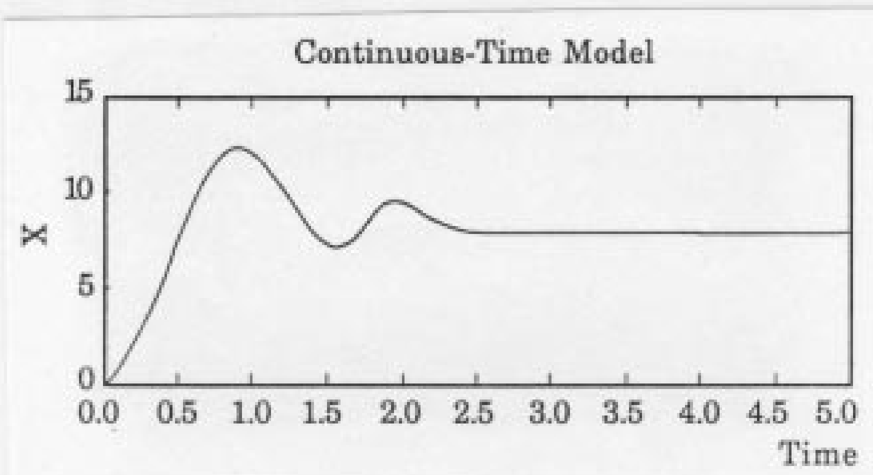

Figure 1.1. Trajectory behavior of a continuous–time model.

We can give the following definitions for continuous–time models:

> "Continuous–time models are characterized by the fact that, within a finite time span, the state variables change their values infinitely often."

No other mathematical model shares this property.

Continuous–time models are represented through sets of differential equations. Among the continuous–time models, two separate classes can be distinguished: the *lumped parameter models* which are described by *ordinary differential equations* (ODE's), in general:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{1.1}$$

and for the special case of *linear systems*:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{1.2}$$

and the *distributed parameter models* which are described by *partial differential equations* (PDE's) such as the diffusion equation:

$$\frac{\partial u}{\partial t} = \sigma \cdot \frac{\partial^2 u}{\partial x^2} \tag{1.3}$$

Both types will be encountered in this text, and it is indeed the set of the continuous–time models that is at the center of our interest.

The second class of mathematical models to be mentioned is the set of *discrete–time models*. Fig.1.2 depicts the trajectory behavior exhibited by discrete–time models.

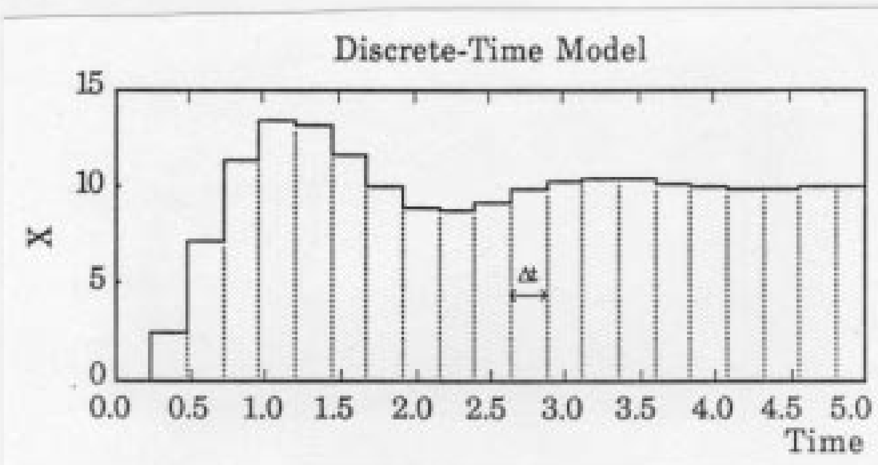

Figure 1.2. Trajectory behavior of a discrete–time model.

In these type of models, the time axis is discretized. Discrete–time models are commonly represented through sets of *difference equations*, at least if the discretization is equidistantly spaced. Such models can be represented as:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) \tag{1.4}$$

In the case of non–equidistantly spaced discrete–time models, a *discrete–event representation* is generally preferred (cf. later).

Discrete–time models can occur naturally. For example, the population dynamics of an insect population is commonly represented through a set of difference equations, since the insects breed only during a short period of the year, i.e., a discretization interval of one year is natural.

Discrete–time models occur frequently in engineering systems, most commonly in *computer–controlled systems*. If a digital computer is used in a control system to compute one or several control signals, it cannot do so on a continuous basis since the algorithm to compute the next set of values of the control signals requires time. It is therefore most natural to apply "time–slicing", i.e., to cut the time axis into short and equidistant intervals where each interval is usually chosen to be just long enough to allow the digital computer to compute one new set of values. If the system to be controlled is itself a continuous–time system (as is often the case), we call this a *sampled–data control system*.

Discrete–time models can also be discretized versions of continuous time models. This is in fact very common. For instance, if we discretize the time axis of the continuous–time state–space model:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$

with a discretization interval Δt, the state derivative becomes:

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} \approx \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) \tag{1.5}$$

or:

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + \Delta t \cdot \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) \tag{1.6}$$

which immediately leads us to a discrete–time model. In fact, whenever we use a digital computer to simulate a continuous–time model, we actually *must* discretize the time axis in some way in order to avoid the problem with the infinitely many state changes in a finite time span. However, any garden variety simulation language (such as ACSL or DARE–P) will do this discretization for us, and hide the fact from us ... except when something goes wrong. Then, we must understand how the program works in order to be able to help it get back on track. This will be one of the major topics to be discussed in the second volume of this text.

The third class of models is the set of *qualitative models* which are by nature discrete–time models (although not necessarily with equidistant time–slicing). Fig.1.3 shows the trajectory behavior of a qualitative model:

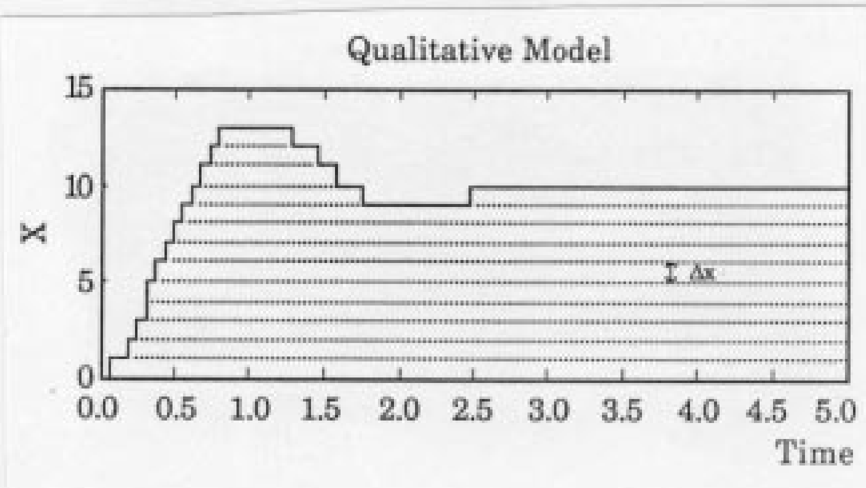

Figure 1.3. Trajectory behavior of a qualitative model.

In a qualitative model, the dependent variables are discretized. Qualitative models are usually coded using a *finite state representation*, and also this type of model will be covered in the text.

The fourth and final class of models is the set of *discrete–event models*. Paradoxically, both the time axis and the state axis of discrete–event models are usually "continuous" (i.e., *real* rather than *integer*), but discrete–event models differ from the continuous–time models by the fact that, in a finite time span, only a finite number of state changes may occur. Fig.1.4 depicts the typical trajectory behavior of a state variable in a discrete–event simulation.

Discrete–event models have been the main topic of a series of previous simulation texts [1.8,1.9,1.10], and will not be covered here.

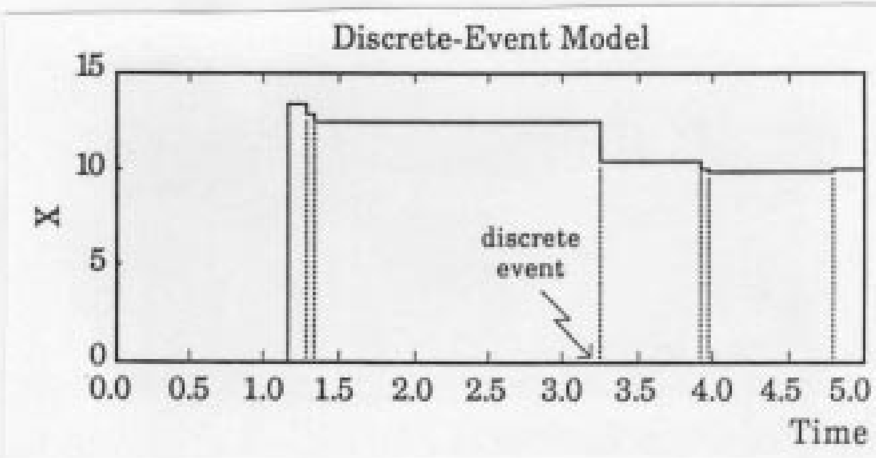

Figure 1.4. Trajectory behavior of a discrete–event model.

When should we use what type of model? Walter Karplus generated a "rainbow" (the way children draw it) that answers this question in a systematic way [1.4]. Fig.1.5 represents a slightly modified version of that "rainbow".

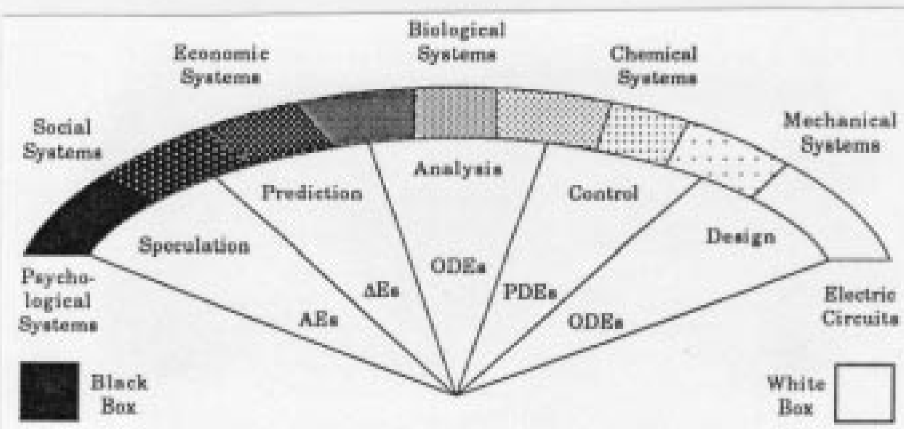

Figure 1.5. Spectrum of modeling and simulation.

Above the rainbow, various application areas of modeling and simulation are shown. They range from electrical circuits to psychological systems. The application areas shown are exemplary. Areas that are not shown include the thermal, hydraulic, and pneumatic systems which should be located somewhere between the mechanical and the chemical systems. In this text, we shall proceed along the rainbow from the right to the left, i.e., from well–defined ("white box") systems to ill–defined ("black box") systems.

Immediately below the rainbow, common purposes for modeling and simulation are specified. Remember that modeling and simulation are always goal–driven, i.e., we should know the purpose of our potential model before we sit down to create it.

Electrical circuits are so well understood that it is possible to use a model to *design* an overall circuit, i.e., once the performance of the model is satisfactory, we can build the real system, and, in all likelihood, it will work just as predicted by the model. This is also true for some of the mechanical systems (except where non-linearities and friction effects become dominant factors).

This is, however, no longer true for chemical systems. Many factors influence a chemical reaction, factors which are all of approximately equal importance. Therefore, models that are valid for a large set of experiments cannot be specified. Thus, a theoretically derived model of a chemical process may predict one thing while the real system that is built after the model may react quite differently. Yet, if we build the system first and match the model to the system, the model contains sufficient internal validity to allow us to build a model of a *controller* for that system which, when applied to the real system, may still work nicely. This is due to the fact that feedback controllers have a tendency to reduce the system's sensitivity to parameter variations.

When we proceed further to the left, we find that the internal validity of our models decays further and further. Eventually, we come to a point where the available models no longer contain sufficient internal validity to allow us to use the model for any design purposes. Yet, we can still use the model for *analyzing* the system behavior, i.e., the internal structure of the model is still sufficiently valid to allow us to reason competently about cause–effect relationships among the variables that are captured by the model.

Advancing further to the left, we come to a point where even this statement is no longer true. Such models are constructed in a mostly inductive manner, and a decent match between model and system behavior no longer guarantees that the internal structure of the model represents the internal structure of the real system in any meaningful way. Yet, we may still be able to *predict* the future of the real system from simulating the model beyond the current time.

Finally, systems exist where even this is no longer true. All we can achieve is to *speculate* about possible futures, maybe with probability tags attached to the various possible outcomes. This is true in particular for social and psychological systems since they are retroactive. These systems include humans who, due to their knowledge of the model predictions, will adjust their behavior to modify that same outcome. In some cases, we end up with self–fulfilling prophecy. If I have a "good" model of the stock market which predicts the growth of a particular stock, and if many people have access to that model and believe in its value, then all these people will wish to buy that particular stock, and sure enough, the stock will gain value (at least for a while). The opposite can also occur. If my model predicts a major disaster, and if a sufficiently large number of influential people know about that prediction and believe in the accuracy of my model, they will do their best to modify the system behavior to prevent that very disaster from ever happening. Good examples are George Orwell's book *1984* and Jay Forrester's world model which predicted clearly undesirable futures. Consequently, legislative actions were taken that hopefully will prevent those very predictions from ever becoming a reality. Walter Karplus wrote rightly that the major purpose of such models is to "arouse public opinion" [1.4].

Below the purpose spectrum, a tool spectrum is presented. Electrical circuits can be accurately described by *ordinary differential equations* (ODE's), since the influence of geometry is usually negligible. This is true except for very high frequencies (microwaves), or for very small dimensions (integrated circuits).

When geometry becomes important, we must introduce the space dimensions as additional independent variables, and we end up with distributed parameter models which are described by *partial differential equations* (PDE's). This is true for mechanical systems with finite stiffness, for thermodynamics, fluid dynamics, optics, and diffusion processes in chemistry.

Advancing further to the left, the available data and the limited knowledge of the meta–laws of these systems no longer warrant the specification of distributed parameter models, and we use again ODE's, not because that is how these systems really behave, but because we cannot validate any more complex models with our limited understanding of the processes, and with the limited experimental data available.

When even less information is present, the accuracy that ODE's provide (and that we must pay for in terms of computing time) is no longer warranted. It makes sense to use very high order integration algorithms only for the best–defined systems, such as those in celestial mechanics. When we simulate a celestial mechanics problem, we like to use an eighth order Runge–Kutta algorithm, since it allows us to select a large integration step size, and yet integrates the model equations with high accuracy. Fourth order algorithms are optimal for most engineering tasks. As a rule of thumb, we use a k^{th} order algorithm if we wish to obtain results with an accuracy of k decimals. For systems with an inherent accuracy of several percent (such as in biology), it does not make sense to use any integration algorithm of higher than first order, i.e., the forward Euler algorithm shown in eq(1.6) is appropriate. Such models are therefore often represented in the form of *difference equations* (ΔE's).

Finally, in the "darkest" of all worlds, i.e., in social and psychological modeling, the models used are mostly static. They are described by *algebraic equations* (AE's). They are usually entirely inductive, and depend on "gut feeling" or the position of the stars in the sky.

1.10 Direct Versus Inverse Problems

Envisage a system as depicted in Fig.1.6.

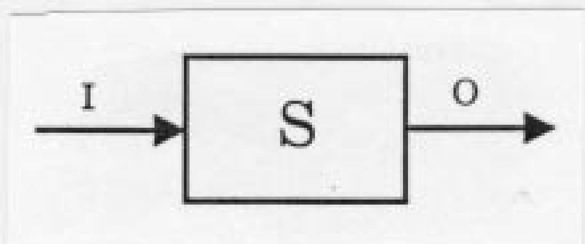

Figure 1.6. Block–diagram of a system.

The system is characterized by a set of *inputs* (I) (including both control inputs and disturbances), by a set of *outputs* (O), and by a set of *internal variables* (S) (including both the state variables and any auxiliary algebraic variables).

The "normal" situation for a simulation is given, when *all* inputs are known as functions over time, and when the system structure and the initial conditions of all state variables are specified. The task of the simulation is to determine the trajectory behavior of *all* outputs, i.e.,

$$\mathbf{I}, \mathbf{S} = \textit{known} \; ; \; \mathbf{O} = \textit{unknown}$$

This problem is called the *direct problem.*

However, two types of inverse problems exist as well. For instance, it could be that the system under study is a "black box". While all inputs and outputs are known, the internal structure of the system and/or the initial values of the state variables are unknown, i.e.,

$$\mathbf{I}, \mathbf{O} = \textit{known} \; ; \; \mathbf{S} = \textit{unknown}$$

These problems are referred to as the *structure identification problem* and the *state estimation problem*, respectively. We shall demonstrate in the second volume of this text how simulation *can* be used to solve identification problems.

A third type of problem is given if:

$$\mathbf{S}, \mathbf{O} = \textit{known} \; ; \; \mathbf{I} = \textit{unknown}$$

This is referred to as the *control problem*, and is the major subject of the area of automatic control. In the second volume, we shall also demonstrate how simulation can be used to solve control problems.

1.11 Summary

In this chapter, we have given some basic definitions, we have outlined the scope of our undertaking, and we have tried to answer the question, why students might be interested in this subject, and why they might want to continue with this course.

References

[1.1] W. Ross Ashby (1956), *An Introduction to Cybernetics*, John Wiley, New York.

[1.2] Brian Gaines (1979), "General Systems Research: Quo Vadis", *General Systems Yearbook*, **24**, pp. 1–9.

[1.3] Thomas Kailath (1980), *Linear Systems*, Information and System Sciences Series, Prentice–Hall, Englewood Cliffs, N.J.

[1.4] Walter J. Karplus (1976), "The Spectrum of Mathematical Modeling and Systems Simulation", Proceedings Eighth AICA Congress on Simulation of Systems, (L. Dekker, ed.), North-Holland Publishing Company, pp. 5–13.

[1.5] Jack P. C. Kleijnen (1982), "Experimentation with Models: Statistical Design and Analysis Techniques", in: *Progress in Modelling and Simulation*, (F.E. Cellier, ed.), Academic Press, London, pp. 173–185.

[1.6] Granino A. Korn, and John V. Wait (1978), *Digital Continuous-System Simulation*, Prentice–Hall, Englewood Cliffs, N.J.

[1.7] Marvin Minsky (1965), "Models, Minds, Machines", Proceedings IFIP Congress, pp. 45–49.

[1.8] A. Alan B. Pritsker (1985), *Introduction to Simulation and SLAM-II*, Third Edition, Halsted Press, New York.

[1.9] Bernard P. Zeigler (1976), *Theory of Modeling and Simulation*, John Wiley, New York.

[1.10] Bernard P. Zeigler (1984), *Multifaceted Modeling and Discrete Event Simulation*, Academic Press, London, U.K.

[1.11] Bernard P. Zeigler, François E. Cellier, and Jerzy W. Rozenblit (1988), "Design of a Simulation Environment for Laboratory Management by Robot Organizations", *J. of Intelligent and Robotic Systems*, **1**, pp. 299–309.

Bibliography

[B1.1] Rutherford Aris (1978), *Mathematical Modelling Techniques*, Pitman, London and San Francisco.

[B1.2] Edward Beltrami (1987), *Mathematics for Dynamic Modeling*, Academic Press, Boston, MA.

[B1.3] Edward A. Bender (1978), *An Introduction to Mathematical Modeling*, John Wiley, New York.

[B1.4] François E. Cellier, ed. (1982), *Progress in Modelling and Simulation*, Academic Press, London, U.K.

[B1.5] Charles M. Close, and Dean K. Frederick (1978), *Modeling and Analysis of Dynamic Systems*, Houghton Mifflin Company, Boston, MA.

[B1.6] Naim A. Kheir, ed. (1988), *Systems Modeling and Computer Simulation*, Marcel Dekker, New York.

[B1.7] William J. Palm, III (1983), *Modeling, Analysis, and Control of Dynamic Systems*, John Wiley, New York.

[B1.8] Jon M. Smith (1987), *Mathematical Modeling and Digital Simulation for Engineers and Scientists*, John Wiley, New York.

[B1.9] Jan A. Spriet, and Ghislain C. Vansteenkiste (1982), *Computer-Aided Modeling and Simulation*, Academic Press, London, U.K.

Homework Problems

[H1.1] Sampled Data System

Given the following linear differential equation system:

$$\dot{x}_1 = x_2 \tag{H1.1a}$$
$$\dot{x}_2 = x_3 \tag{H1.1b}$$
$$\dot{x}_3 = -2x_1 - 3x_2 - 4x_3 \tag{H1.1c}$$
$$y = 7x_1 - 5x_2 \tag{H1.1d}$$

Use the forward Euler integration algorithm to convert this set of differential equations to a set of difference equations. Use a step size of $\Delta t = 0.1$ *sec*.

[H1.2] Signal Types

For the following systems, try to identify the inputs, the outputs, and disturbances (where applicable):

(a) The water level in a reservoir

(b) The power supply of a city

(c) A car being driven along a mountain road

(d) A toaster

[H1.3]* Meta–Models

Given the following non–linear second–order model:

$$\dot{x}_1 = -3x_1 + 1.5x_1x_2 \tag{H1.3a}$$
$$\dot{x}_2 = 4.5x_2 - 1.5x_1x_2 \tag{H1.3b}$$

We want to analyze the behavior of this system (a so–called Lotka–Volterra model) in the vicinity of all of its steady–state points (*hint:* steady–state points are those points in which all derivatives are equal to zero). This task is accomplished through the use of linear meta-models. The problem can be decomposed into the following subtasks:

(a) Determine all steady-state points of this system.

(b) For any steady-state point that is not the origin ($[x_1, x_2] = [0, 0]$) itself, apply a linear variable transformation which moves the steady-state point to the origin, i.e., if $[x_1, x_2]_{ss} = [a, b]$, we introduce the new set of state variables: $\xi_1 = x_1 - a$ and $\xi_2 = x_2 - b$, and rewrite our set of differential equations in terms of these new variables. In the new coordinate system $[\xi_1, \xi_2]$, the steady-state point will be the origin.

(c) We can now linearize the models around their origins by simply throwing out all non-linear terms.

(d) The resulting linear meta models are so simple that they are amenable to an analytical treatment (i.e., they have closed-form solutions). Find these solutions, and determine qualitatively the behavior of the non-linear system in the vicinity of these steady-state points. Sketch a graph of what you expect the trajectories to look like in the $[x_1, x_2]$ plane (the so-called phase plane) in the vicinity of the steady-state points.

Projects

[P1.1] Definitions

Get a number of simulation and/or system theory textbooks from your library, and compile a list of definitions of "What is a System?" Write a term paper in which these definitions are critically reviewed and classified. (Such a compilation has actually been published once.)

Research

[R1.1] Experimental Frames

Study the separation of the model description from the experiment description. Analyze under what conditions such a separation is meaningful and/or feasible. Develop mechanisms to ensure the compatibility of a proposed experiment with a given model (mechanisms to code the experimental frame), and develop a generic language to code the experimental frame specification.