System Dynamics Methods:
A Quick Introduction

Craig W. Kirkwood
Arizona State University

Contents

Preface

These notes provide a quick introduction to system dynamics methods using business examples. The methods of system dynamics are general, but their implementation requires that you use specific computer software. A number of different software packages are available to implement system dynamics, and the Vensim modeling package is used in these notes. This package was selected because i) it supports a compact, but informative, graphical notation, ii) the Vensim equation notation is compact and complete, iii) Vensim provides powerful tools for quickly constructing and analyzing process models, and iv) a version is available free for instructional use over the World Wide Web at http://www.vensim.com. A quick reference and tutorial for Vensim can be downloaded from my system dynamics home page at www.public.asu.edu/∼kirkwood/sysdyn/SDRes.htm.

If you obtained this document in electronic form and wish to print it, please note that it is formatted for two-sided printing. The blank pages at the end of some chapters are intentional so that new chapters will start on right-hand pages.

Special thanks to Robert Eberlein for many helpful comments on drafts of these notes.

CHAPTER 1

System Behavior and Causal Loop Diagrams

Human beings are quick problem solvers. From an evolutionary standpoint, this makes sense—if a sabertooth tiger is bounding toward you, you need to quickly decide on a course of action, or you won't be around for long. Thus, quick problem solvers were the ones who survived. We quickly determine a *cause* for any *event* that we think is a problem. Usually we conclude that the cause is another event. For example, if sales are poor (the event that is a problem), then we may conclude that this is because the sales force is insufficiently motivated (the event that is the cause of the problem).

This approach works well for simple problems, but it works less well as the problems get more complex, for example in addressing management problems which are cross-functional or strategic. General Motors illustrates the issue. For over half a century, GM dominated the automotive industry. GM's difficulties did not come from a lightning attack by Japanese auto manufacturers. GM had a couple of decades to adapt, but today it is still attempting to find a way to its former dominance, more than three decades after the start of Japanese automobile importation. During this period, many of GM's employees and managers have turned over, but the company still has difficulty adjusting. There seems to be something about the way that GM is put together that makes its behavior hard to change.

1.1 Systems Thinking

The methods of *systems thinking* provide us with tools for better understanding these difficult management problems. The methods have been used for over thirty years (Forrester 1961) and are now well established. However, these approaches require a shift in the way we think about the performance of an organization. In particular, they require that we move away from looking at isolated *events* and their *causes* (usually assumed to be some other events), and start to look at the organization as a *system* made up of interacting parts.

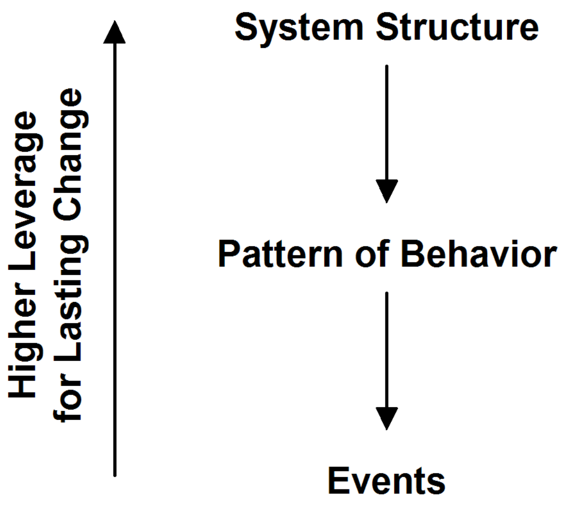


Figure 1.1 *Looking for high leverage*

We use the term *system* to mean an interdependent group of items forming a unified pattern. Since our interest here is in business processes, we will focus on systems of people and technology intended to design, market, produce, and distribute products or services. Almost everything that goes on in business is part of one or more systems. As noted above, when we face a management problem we tend to assume that some external event caused it. With a systems approach, we take an alternative viewpoint—namely that the internal structure of the system is often more important than external events in generating the problem.

This is illustrated by the diagram in Figure 1.1.[1] Many people try to explain business performance by showing how one set of events causes another or, when they study a problem in depth, by showing how a particular set of events is part of a longer term *pattern of behavior*. The difficulty with this "events causes events" orientation is that it doesn't lead to very powerful ways to alter the undesirable performance. This is because you can always find yet another event that caused the one that you thought was the cause. For example, if a new product is not selling (the event that is a problem), then you may conclude that this if because the sales force is not pushing it (the event that is the cause of the problem). However, you can then ask why the sales force is not pushing it (another problem!). You might then conclude that this is because they are overworked (the cause of your new problem). But you can then look for the cause of this condition. You can continue this process almost forever, and thus it is difficult to determine what to do to improve performance.

[1] Figure 1.1 and this discussion of it are based on class notes by John Sterman of the MIT Sloan School of Management.

If you shift from this event orientation to focusing on the internal *system structure*, you improve your possibility of improving business performance. This is because system structure is often the underlying source of the difficulty. Unless you correct system structure deficiencies, it is likely that the problem will resurface, or be replaced by an even more difficult problem.

1.2 Patterns of Behavior

To start to consider system structure, you first generalize from the specific events associated with your problem to considering *patterns of behavior* that characterize the situation. Usually this requires that you investigate how one or more variables of interest change over time. (In a business setting, variables of interest might be such things as cost, sales, revenue, profit, market share, and so forth.) That is, what *patterns of behavior* do these variables display. The systems approach gains much of its power as a problem solving method from the fact that similar patterns of behavior show up in a variety of different situations, and the underlying system structures that cause these characteristic patterns are known. Thus, once you have identified a pattern of behavior that is a problem, you can look for the system structure that is know to cause that pattern. By finding and modifying this system structure, you have the possibility of permanently eliminating the problem pattern of behavior.

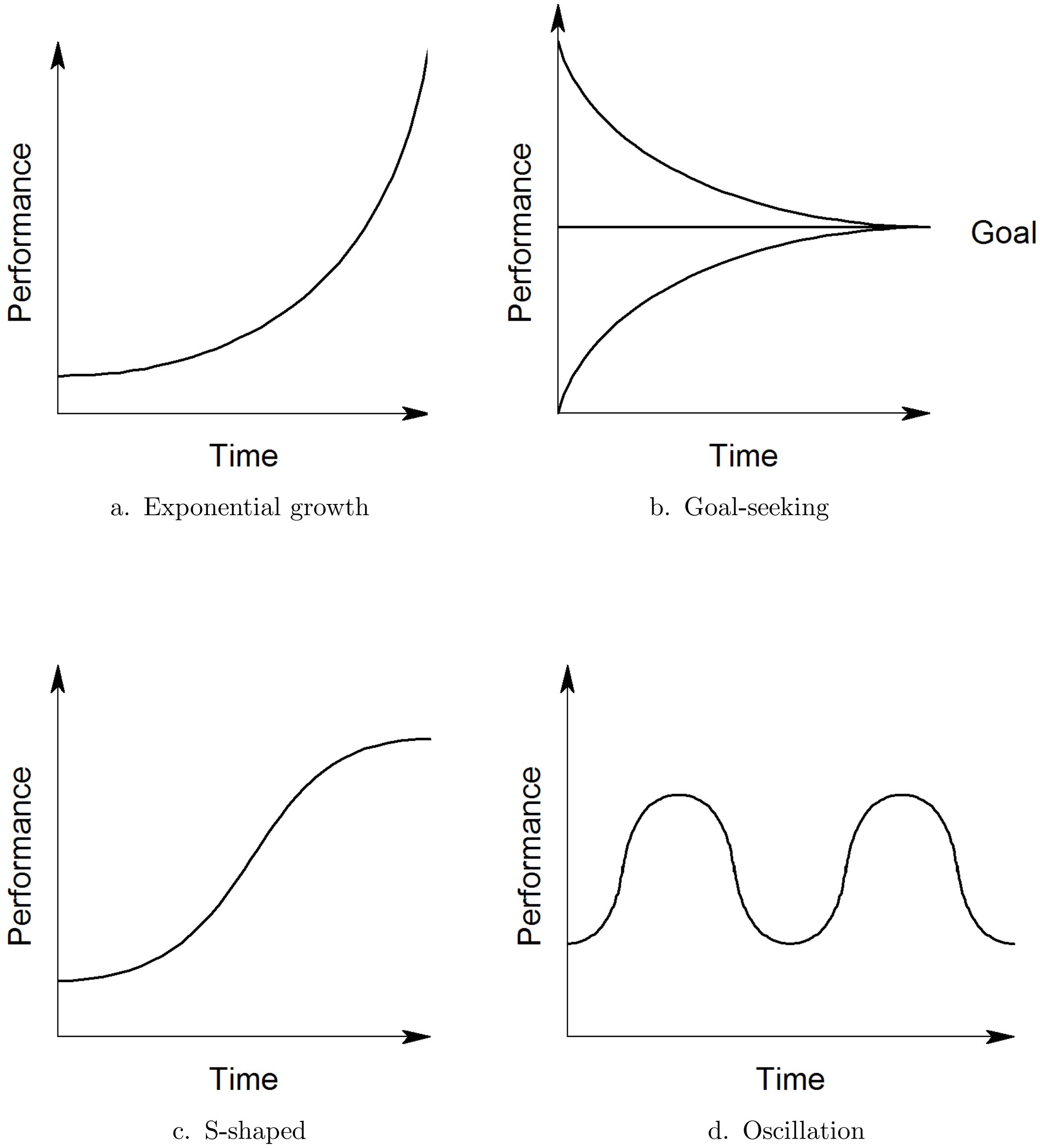
The four patterns of behavior shown in Figure 1.2 often show up, either individually or in combinations, in systems. In this figure, “Performance” refers to some variable of interest. This is often a measure of financial or operational effectiveness or efficiency. In this section, we summarize the characteristics of these patterns. In later sections, we examine the types of system structures which generate these patterns.[2]

With **exponential growth** (Figure 1.2a), an initial quantity of something starts to grow, and the rate of growth increases. The term “exponential growth” comes from a mathematical model for this increasing growth process where the growth follows a particular functional form called the exponential. In business processes, the growth may not follow this form exactly, but the basic idea of accelerating growth holds. This behavior is what we would like to see for sales of a new product, although more often sales follow the s-shaped curve discussed below.

With **goal-seeking** behavior (Figure 1.2b), the quantity of interest starts either above or below a goal level and over time moves toward the goal. Figure 1.2b shows two possible cases, one where the initial value of the quantity is above the goal, and one where the initial value is below the goal.

With **s-shaped growth** (Figure 1.2c), initial exponential growth is followed by goal-seeking behavior which results in the variable leveling off.

[2] The following discussion draws on Senge (1990), Senge et al (1994), and notes from David Kreutzer and John Sterman.



a. Exponential growth

b. Goal-seeking

c. S-shaped

d. Oscillation

Figure 1.2 *Characteristic patterns of system behavior*

With **oscillation** (Figure 1.2d), the quantity of interest fluctuates around some level. Note that oscillation initially appears to be exponential growth, and then it appears to be s-shaped growth before reversing direction.

Common combinations of these four patterns include

- Exponential growth combined with oscillation. With this pattern, the general trend is upward, but there can be declining portions, also. If the magnitude of the oscillations is relatively small, then growth may plateau, rather than actually decline, before it continues upward.
- Goal-seeking behavior combined with an oscillation whose amplitude gradually declines over time. With this behavior, the quantity of interest will overshoot the goal on first one side and then the other. The amplitude of these overshoots declines until the quantity finally stabilizes at the goal.
- S-shaped growth combined with an oscillation whose amplitude gradually declines over time.

1.3 Feedback and Causal Loop Diagrams

To better understand the system structures which cause the patterns of behavior discussed in the preceding section, we introduce a notation for representing system structures. The usefulness of a graphical notation for representing system structure is illustrated by the diagram in Figure 1.3 which is adapted from a figure in Richardson and Pugh (1981). This shows the relationships among the elements of a production sector within a company. In this diagram, the short descriptive phrases represent the elements which make up the sector, and the arrows represent the causal influences between these elements. For example, examining the left hand side of the diagram, we see that "Production" is directly influenced by "Workforce (production capacity)" and "Productivity." In turn, "Production" influences "Receipt into inventory."

This diagram presents relationships that are difficult to verbally describe because normal language presents interrelations in linear cause-and-effect chains, while the diagram shows that in the actual system there are circular chains of cause-and-effect. Consider, for example, the "Inventory" element in the upper left-hand corner of the diagram. We see from the diagram that "Inventory" influences "Availability of inventory," which in turn influences "Shipments." To this point in the analysis, there has been a linear chain of cause and effect, but continuing in the diagram, we see that "Shipments" influence "Inventory." That is, the chain of causes and effects forms a closed loop, with "Inventory" influencing itself indirectly through the other elements in the loop. The diagram shows this more easily than a verbal description.

When an element of a system indirectly influences itself in the way discussed for Inventory in the preceding paragraph, the portion of the system involved is called a *feedback loop* or a *causal loop*. [Feedback is defined as the transmission and return of information (Richardson and Pugh 1981).] More formally, a feedback loop is a *closed sequence of causes and effects, that is, a closed path of*

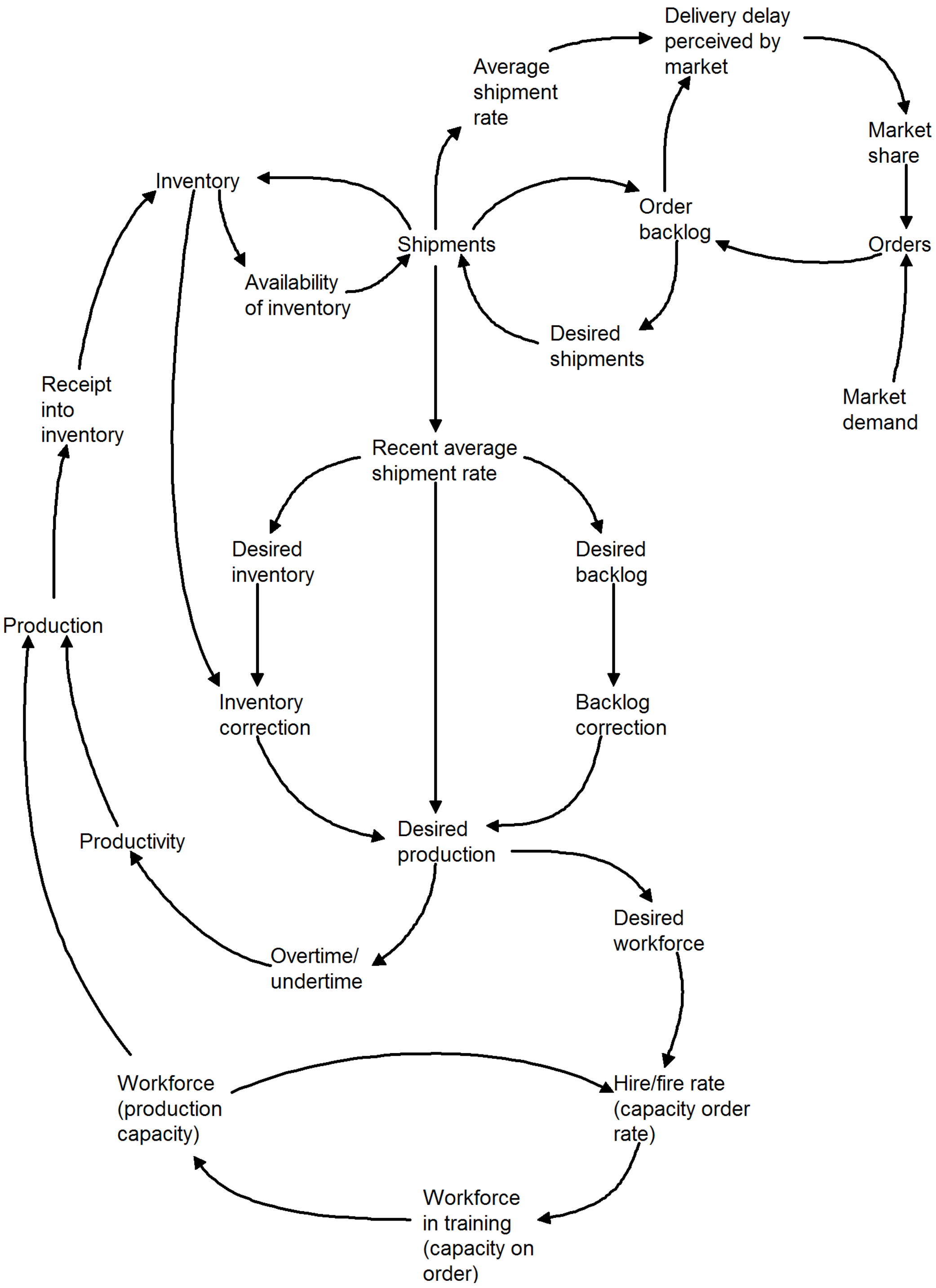


Figure 1.3 *Feedback structure of a basic production sector*

action and information (Richardson and Pugh 1981). The reason for emphasizing feedback is that it is often necessary to consider feedback within management systems to understand what is causing the patterns of behavior discussed in the preceding section and shown in Figure 1.2. That is, the causes of an observed pattern of behavior are often found within the feedback structures for a management system.

To complete our presentation of terminology for describing system structure, note that a linear chain of causes and effects which does not close back on itself is called an *open loop*. An analysis of causes and effects which does not take into account feedback loops is sometimes called *open loop thinking*, and this term usually has a pejorative connotation—it indicates thinking that is not taking the full range of impacts of a proposed action into account.

A map of the feedback structure of a management system, such as that shown in Figure 1.3, is a starting point for analyzing what is causing a particular pattern of behavior. However, additional information aids with a more complete analysis. Figure 1.4 defines notation for this additional information. This figure is an annotated *causal loop diagram* for a simple process, filling a glass of water. This diagram includes *elements* and arrows (which are called *causal links*) linking these elements together in the same manner as shown in Figure 1.3, but it also includes a sign (either + or −) on each link. These signs have the following meanings:

1 A causal link from one element A to another element B is *positive* (that is, +) if either (a) A adds to B or (b) a change in A produces a change in B in the *same* direction.

2 A causal link from one element A to another element B is *negative* (that is, −) if either (a) A subtracts from B or (b) a change in A produces a change in B in the *opposite* direction.

This notation is illustrated by the causal loop diagram in Figure 1.4. Start from the element "Faucet Position" at the top of the diagram. If the faucet position is increased (that is, the faucet is opened further) then the "Water Flow" increases. Therefore, the sign on the link from "Faucet Position" to "Water Flow" is positive. Similarly, if the "Water Flow" increases, then the "Water Level" in the glass will increase. Therefore, the sign on the link between these two elements is positive.

The next element along the chain of causal influences is the "Gap," which is the difference between the "Desired Water Level" and the (actual) "Water Level." (That is, Gap = Desired Water Level − Water Level.) From this definition, it follows that an increase in "Water Level" decreases "Gap," and therefore the sign on the link between these two elements is negative. Finally, to close the causal loop back to "Faucet Position," a greater value for "Gap" presumably leads to an increase in "Faucet Position" (as you attempt to fill the glass) and therefore the sign on the link between these two elements is positive. There is one additional link in this diagram, from "Desired Water Level" to "Gap." From

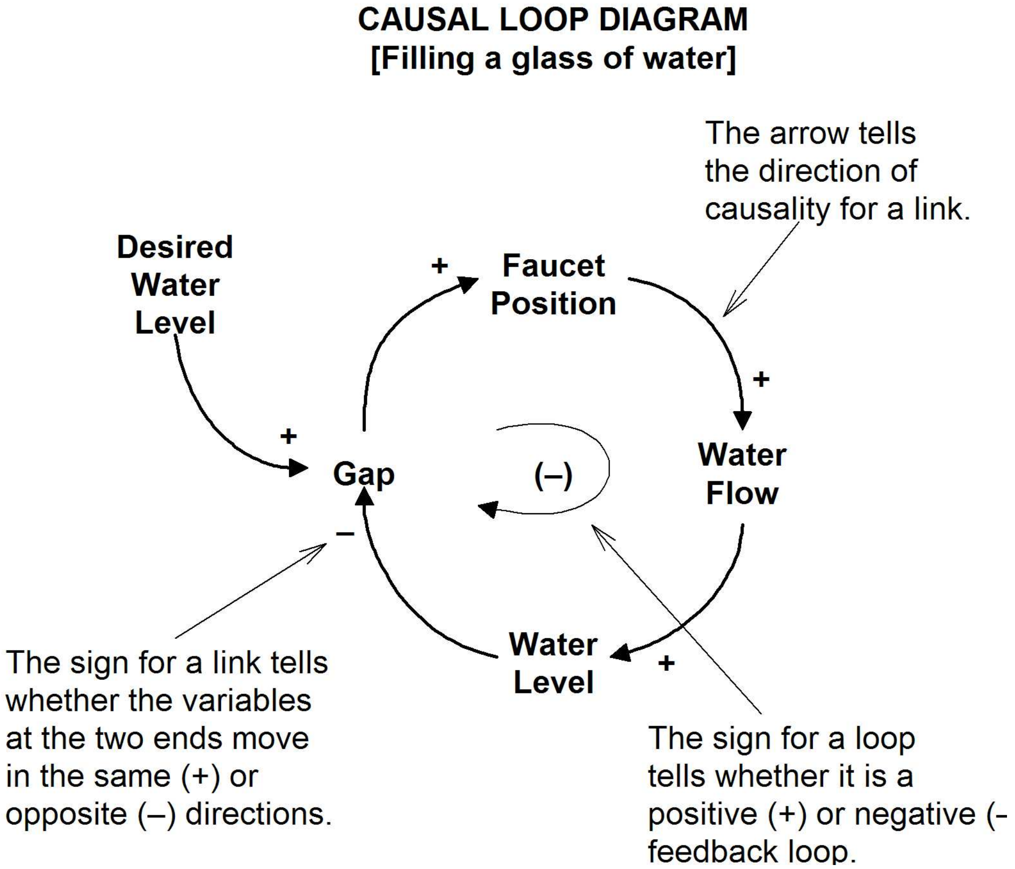

CAUSAL LOOP DIAGRAM
[Filling a glass of water]
The arrow tells the direction of causality for a link.
Desired Water Level
+
Faucet Position
+
+
Gap
(–)
Water Flow
–
Water Level
+
The sign for a link tells whether the variables at the two ends move in the same (+) or opposite (–) directions.
The sign for a loop tells whether it is a positive (+) or negative (- feedback loop.

Figure 1.4 *Causal loop diagram notation*

the definition of "Gap" given above, the influence is in the same direction along this link, and therefore the sign on the link is positive.

In addition to the signs on each link, a complete loop also is given a sign. The sign for a particular loop is determined by counting the number of minus (−) signs on all the links that make up the loop. Specifically,

1 A feedback loop is called *positive*, indicated by a + sign in parentheses, if it contains an even number of negative causal links.

2 A feedback loop is called *negative*, indicated by a − sign in parentheses, if it contains an odd number of negative causal links.

Thus, the sign of a loop is the algebraic product of the signs of its links. Often a small looping arrow is drawn around the feedback loop sign to more clearly indicate that the sign refers to the loop, as is done in Figure 1.4. Note that in this diagram there is a single feedback (causal) loop, and that this loop has one negative sign on its links. Since one is an odd number, the entire loop is negative.

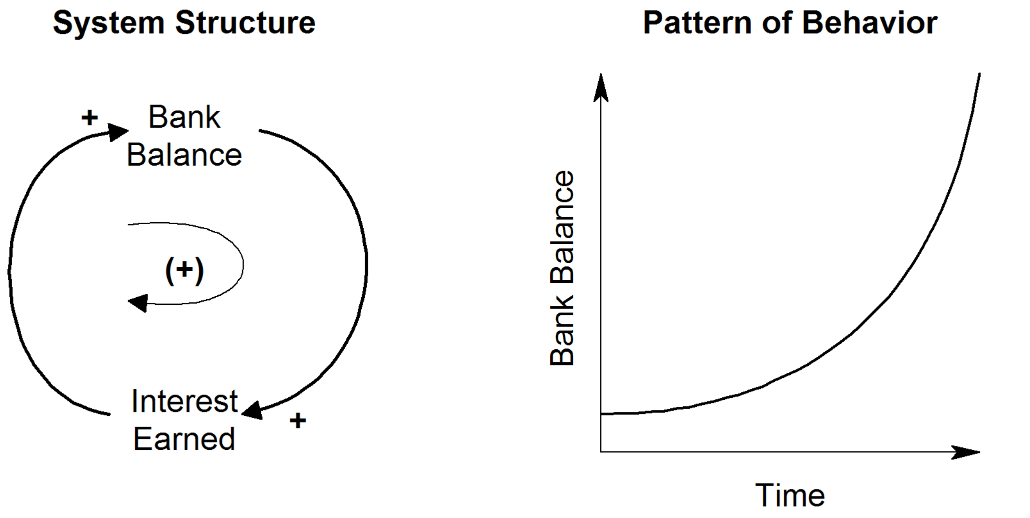


Figure 1.5 *Positive (reinforcing) feedback loop: Growth of bank balance*

An alternative notation is used in some presentations of causal loop diagrams. With this alternate notation, a lower case s is used instead of a + on a link, and a lower case o is used instead of a −. The s stands for "same," and the o stands for "opposite," indicating that the variables at the two ends of the link move in either the same direction (s) or opposite directions (o). For the loops, a capital R is used instead of (+), and a capital B is used instead of (−). The R stands of "reinforcing," and the B stands for "balancing." The reason for using these specific terms will become clearer as we discuss the patterns of behavior associated with different system structures in the next section.

1.4 System Structure and Patterns of Behavior

This section presents simple structures which lead to the typical patterns of behavior shown earlier in Figure 1.2. While the structures of most management systems are more complicated than those shown here, these structures are building blocks from which more complex models can be constructed.

Positive (Reinforcing) Feedback Loop

A positive, or reinforcing, feedback loop reinforces change with even more change. This can lead to rapid growth at an ever-increasing rate. This type of growth pattern is often referred to as *exponential growth.* Note that in the early stages of the growth, it seems to be slow, but then it speeds up. Thus, the nature of the growth in a management system that has a positive feedback loop can be deceptive. If you are in the early stages of an exponential growth process, something that is going to be a major problem can seem minor because it is

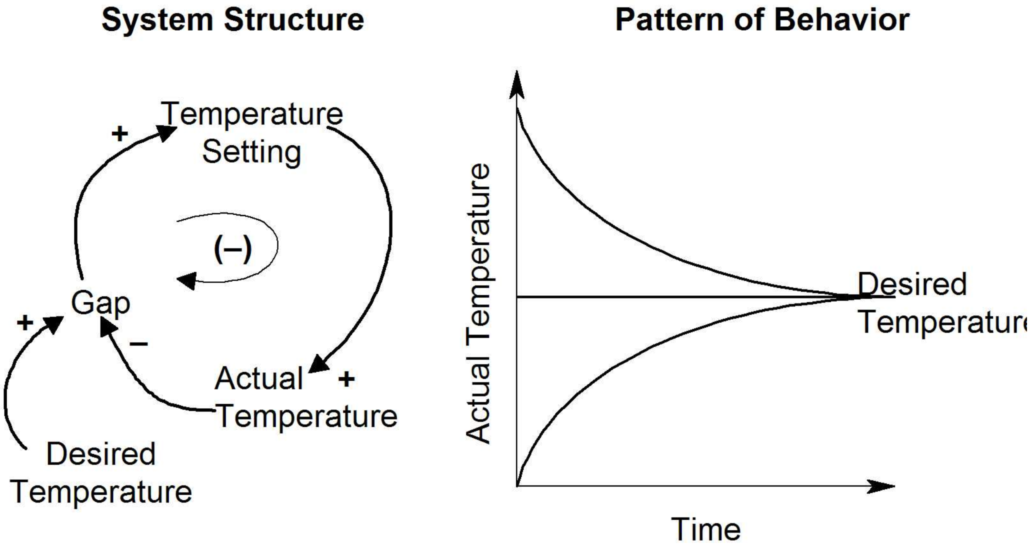


Figure 1.6 *Negative (balancing) feedback loop: Regulating an elective blanket*

growing slowly. By the time the growth speeds up, it may be too late to solve whatever problem this growth is creating. Examples that some people believe fit this category include pollution and population growth. Figure 1.5 shows a well know example of a positive feedback loop: Growth of a bank balance when interest is left to accumulate.

Sometimes positive feedback loops are called vicious or virtuous cycles, depending on the nature of the change that is occurring. Other terms used to describe this type of behavior include bandwagon effects or snowballing.

Negative (Balancing) Feedback Loop

A negative, or balancing, feedback loop seeks a goal. If the current level of the variable of interest is above the goal, then the loop structure pushes its value down, while if the current level is below the goal, the loop structure pushes its value up. Many management processes contain negative feedback loops which provide useful stability, but which can also resist needed changes. In the face of an external environment which dictates that an organization needs to change, it continues on with similar behavior. These types of feedback loops are so powerful in some organizations that the organizations will go out of business rather than change. Figure 1.6 shows a negative feedback loop diagram for the regulation of an electric blanket temperature.

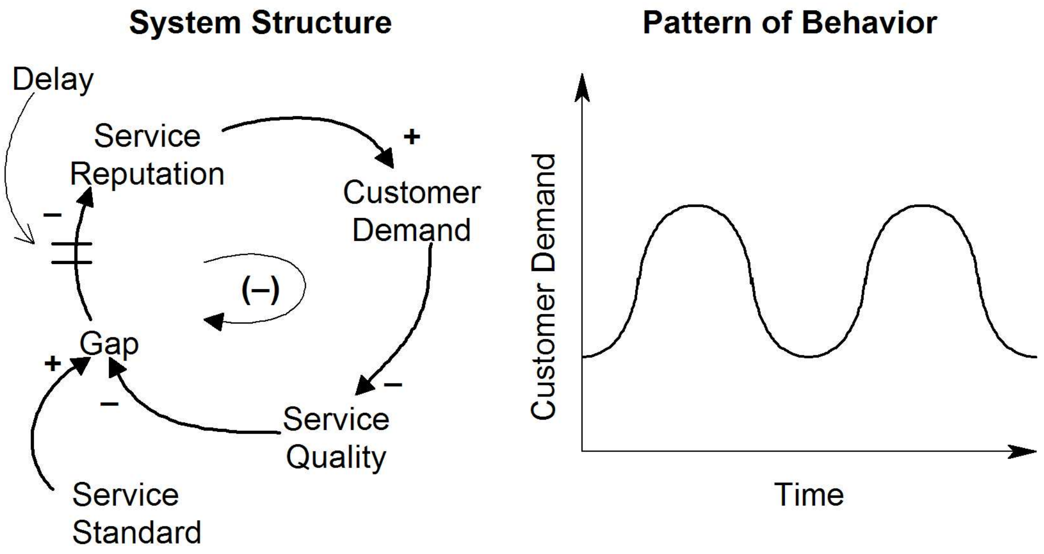


Figure 1.7 *Negative feedback loop with delay: Service quality*

Negative Feedback Loop with Delay

A negative feedback loop with a substantial delay can lead to oscillation. The specific behavior depends on the characteristics of the particular loop. In some cases, the value of a variable continues to oscillate indefinitely, as shown above. In other cases, the amplitude of the oscillations will gradually decrease, and the variable of interest will settle toward a goal. Figure 1.7 illustrates negative feedback with a delay in the context of service quality. (This example assumes that there are fixed resources assigned to service.)

Multi-level production and distribution systems can display this type of behavior because of delays in conveying information about the actual customer demand for a product to the manufacturing facility. Because of these delays, production continues long after enough product has been manufactured to meet demand. Then production is cut back far below what is needed to replace items that are sold while the excess inventory in the system is worked off. This cycle can continue indefinitely, which places significant strains on the management of the process. For example, there may be a pattern of periodic hiring and layoffs. There is some evidence that what are viewed as seasonal variations in customer demand in some industries are actually oscillations caused by delays in negative feedback loops within the production-distribution system.

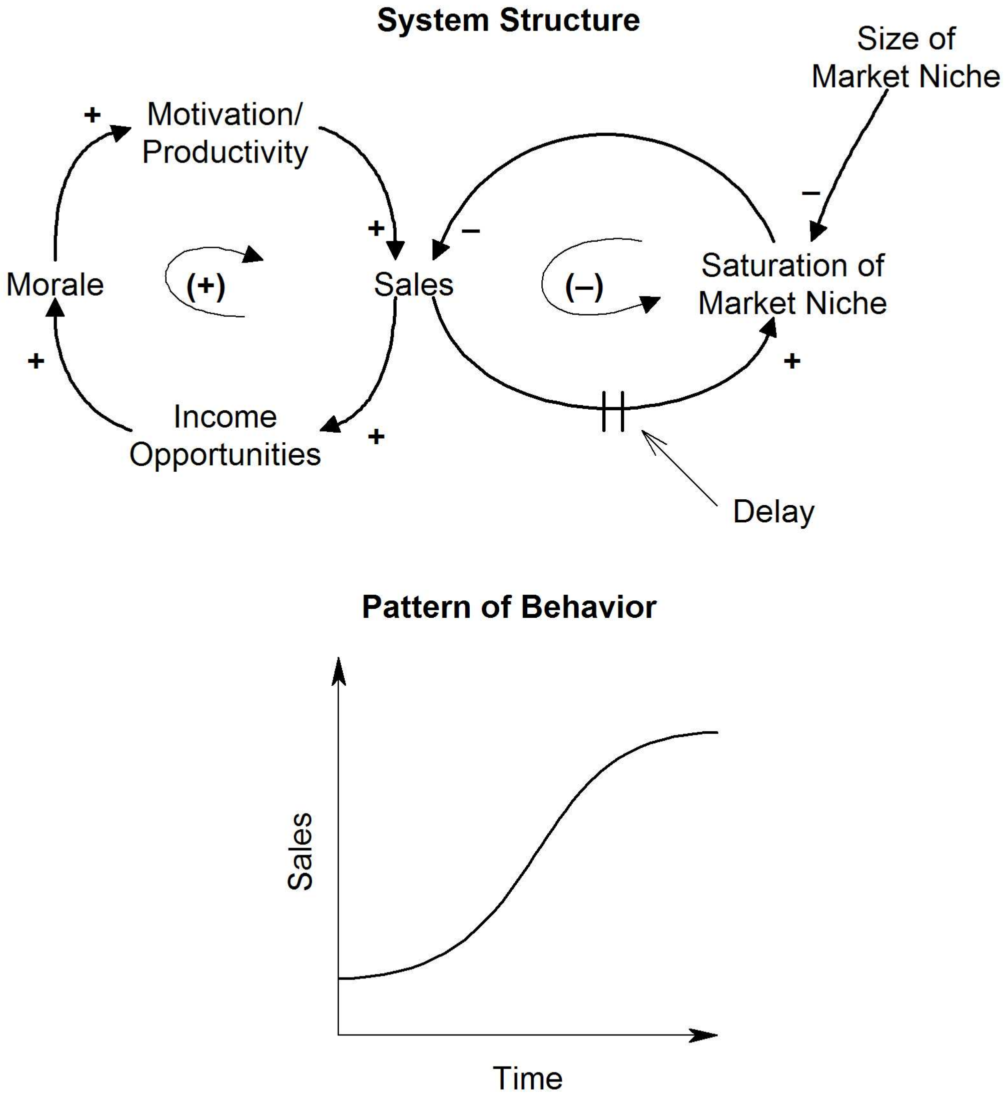


Figure 1.8 *Combination of positive and negative loops: Sales growth*

Combination of Positive and Negative Loops

When positive and negative loops are combined, a variety of patterns are possible. The example above shows a situation where a positive feedback loop leads to early exponential growth, but then, after a delay, a negative feedback loop comes to dominate the behavior of the system. This combination results in an s-shaped pattern because the positive feedback loop leads to initial exponential growth, and then when the negative feedback loop takes over it leads to goal seeking behavior. Figure 1.8 illustrates a combination of positive and negative loops in the context of sales for a new product.

Most growth processes have limits on their growth. At some point, some resource limit will stop the growth. As Figure 1.8 illustrates, growth of sales for a new product will ultimately be slowed by some factor. In this example, the limiting factor is the lack of additional customers who could use the product.

1.5 Creating Causal Loop Diagrams

To start drawing a causal loop diagram, decide which *events* are of interest in developing a better understanding of system structure. For example, perhaps sales of some key product were lower than expected last month. From these events, move to showing (perhaps only qualitatively) the *pattern of behavior* over time for the quantities of interest. For the sales example, what has been the pattern of sales over the time frame of interest? Have sales been growing? Oscillating? S-shaped? Finally, once the pattern of behavior is determined, use the concepts of positive and negative feedback loops, with their associated generic patterns of behavior, to begin constructing a causal loop diagram which will explain the observed pattern of behavior.

The following hints for drawing causal loop diagrams are based on guidelines by Richardson and Pugh (1981) and Kim (1992):

1 Think of the elements in a causal loop diagram as *variables* which can go up or down, but don't worry if you cannot readily think of existing measuring scales for these variables.

- Use nouns or noun phrases to represent the elements, rather than verbs. That is, the actions in a causal loop diagram are represented by the links (arrows), and not by the elements. For example, use "cost" and not "increasing cost" as an element.
- Be sure that the definition of an element makes it clear which direction is "up" for the variable. For example, use "tolerance for crime" rather than "attitude toward crime."
- Generally it is clearer if you use an element name for which the positive sense is preferable. For example, use "Growth" rather than "Contraction."
- Causal links should imply a direction of causation, and not simply a time sequence. That is, a positive link from element A to element B does not mean "first A occurs and then B occurs." Rather it means, "when A increases then B increases."

2 As you construct links in your diagram, think about possible unexpected side effects which might occur in addition to the influences you are drawing. As you identify these, decide whether links should be added to represent these side effects.

3 For negative feedback loops, there is a goal. It is usually clearer if this goal is explicitly shown along with the "gap" that is driving the loop toward the goal. This is illustrated by the examples in the preceding section on regulating electric blanket temperature and service quality.

4 A difference between actual and perceived states of a process can often be important in explaining patterns of behavior. Thus, it may be important to include causal loop elements for both the actual value of a variable and the perceived value. In many cases, there is a lag (delay) before the actual state is perceived. For example, when there is a change in actual product quality, it usually takes a while before customers perceive this change.

5 There are often differences between short term and long term consequences of actions, and these may need to be distinguished with different loops. For example, the short term result of taking a mood altering drug may be to feel better, but the long run result may be addiction and deterioration in health.

6 If a link between two elements needs a lot of explaining, you probably need to add intermediate elements between the two existing elements that will more clearly specify what is happening.

7 Keep the diagram as simple as possible, subject to the earlier points. The purpose of the diagram is not to describe every detail of the management process, but to show those aspects of the feedback structure which lead to the observed pattern of behavior.

1.6 References

J. W. Forrester, *Industrial Dynamics*, The MIT Press, Cambridge, Massachusetts, 1961.

D. H. Kim, "Toolbox: Guidelines for Drawing Causal Loop Diagrams," *The Systems Thinker*, Vol. 3, No. 1, pp. 5–6 (February 1992).

G. P. Richardson and A. L. Pugh III, *Introduction to System Dynamics Modeling with DYNAMO*, Productivity Press, Cambridge, Massachusetts, 1981.

P. M. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday Currency, New York, 1990.

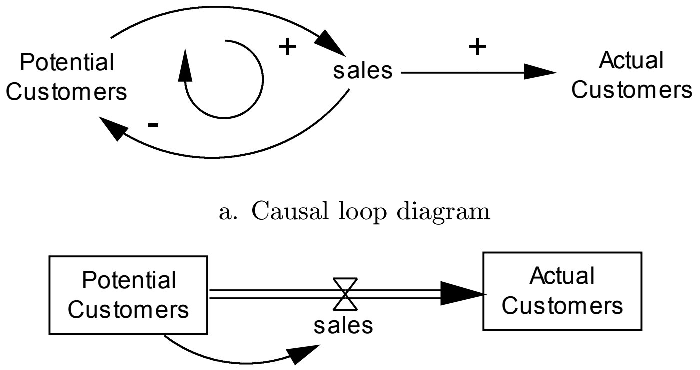
P. M. Senge, C. Roberts, R. B. Ross, B. J. Smith, and A. Kleiner, *The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization*, Doubleday Currency, New York, 1994.

CHAPTER 2

A Modeling Approach

The issues we will address to improve our understanding of how business processes work are illustrated by the causal loop diagram in Figure 2.1a. This models a simple advertising situation for a durable good. There is a pool of Potential Customers who are turned into Actual Customers by sales. Potential Customers and sales are connected in a negative feedback loop with the goal of driving Potential Customers to zero. If we visualize a typical mass advertising situation, we would expect that the greater the number of Potential Customers, the greater the sales, and this is shown in Figure 2.1a by the positive arrow between Potential Customers and sales. Similarly, greater sales lead to fewer Potential Customers (since the Potential Customers are converted into Actual Customers by sales), and hence there is a negative arrow from sales to Potential Customers. Since there are a odd number of negative links in the feedback loop between Potential Customers and sales, this is a negative feedback loop.

We obtain from this diagram the (not very profound) insight that eventually sales must go to zero when the number of Potential Customers reaches zero. However, this insight by itself is not particularly useful for business management purposes because there is no information about the *rate* at which Potential Customers will go to zero. It can make a big difference for managing the production and sales of this product if it will sell well for ten months or ten years before we run out of Potential Customers! For a simple situation like this, we could use a spreadsheet to develop a quantitative model to investigate the rate at which Potential Customers will go to zero, but as the complexity of the situation increases, this becomes more difficult. In the remainder of this chapter, we develop a systematic approach to investigating questions of this type which can be applied to both simple and complex business processes.



a. Causal loop diagram

b. Stock and flow diagram

Figure 2.1 *Advertising example*

2.1 Stock and Flow Diagrams

Figure 2.1b illustrates a graphical notation that provides some structure for thinking about the rate at which Potential Customers goes to zero. This notation consists of three different types of elements: stocks, flows, and information. As we will see below, it is a remarkable fact that the three elements in this diagram provide a general way of graphically representing *any* business process. Furthermore, this graphical notation can be used as a basis for developing a quantitative model which can be used to study the characteristics of the process.

This type of diagram is called a *stock and flow* diagram. As with a causal loop diagram, the stock and flow diagram shows relationships among *variables* which have the potential to change over time. In the Figure 2.1b stock and flow diagram, the variables are Potential Customers, sales, and Actual Customers. Unlike a causal loop diagram, a stock and flow diagram distinguishes between different types of variables. Figure 2.1b shows two different types of variables, which are distinguished by different graphical symbols. The variables Potential Customers and Actual Customers are shown inside rectangles, and this type of variable is called a *stock*, *level*, or *accumulation*. The variable sales is shown next to a bow tie or butterfly valve symbol, and this type of variable is called a *flow*, or *rate*.

To understand and construct stock and flow diagrams, it is necessary to understand the difference between stocks and flows. However, before considering this in more detail, it is useful to discuss what we are attempting to do with this approach to modeling business processes.

2.2 Generality of the Approach

I noted above that the stock and flow notation illustrated in Figure 2.1b provides a general way to graphically characterize any business process. This may seem ambitious: *any* process! In particular, if you have previously worked with computer simulation packages for, to take a specific example, manufacturing processes, you know that they generally contain many more elements than the two shown here. For example, a manufacturing simulation package might contain specific symbols and characterizations for a variety of different milling machines or other manufacturing equipment.

This type of detailed information is important for studying the specific detailed operation of a particular manufacturing process. We will not be providing such details here because they are specific to particular equipment (which will probably soon be obsolete). Instead, we are considering the characteristics that are generally shared by *all* business processes and the components which make up these processes. It is a remarkable fact that all such processes can be characterized in terms of variables of two types, stocks (levels, accumulations) and flows (rates).

The conclusion in the previous paragraph is supported by over a century of theoretical and practical work. Forrester (1961) first systematically applied these ideas to business process analysis almost forty years ago, and extensive practical applications have shown that this way of considering business processes provides significant insights based on solid theory. As the old saying goes, there is nothing more practical than a good theory, and the theory presented here can be turned in practice, yielding competitive advantage.

2.3 Stocks and Flows

The graphical notation in Figure 2.1b hints at the differences between stocks and flows. The rectangular boxes around the variables Potential Customers and Actual Customers look like containers of some sort, or perhaps even bathtubs. The double-line arrow pointing from Potential Customers toward Actual Customers looks like a pipe, and the butterfly valve in the middle of this pipe looks like a valve controlling the flow through the pipe. Thus, the graphical notation hints at the idea that there is a flow from Potential Customers toward Actual Customers, with the rate of the flow controlled by the sales valve. And, in fact, this is the key idea behind the difference between a stock and a flow: A stock is an accumulation of something, and a flow is the movement or flow of the something from one stock to another.

A primary interest of business managers is changes in variables like Actual Customers over time. If nothing changes, then anybody can manage just do what has always been done. Some of the greatest management challenges come from change. If sales start to decline, or even increase, you should investigate why this change has occurred and how to address it. One of the key differences between managers who are successful and those who are not is their ability to address changes before it is too late.

We will focus on investigating these changes, and in particular learning how the elements and structure of a business process can bring about such changes. Because of this focus on the elements which make up a process (which are often referred to as the *components* of a *system*) and how the performance of the process changes over time, the ideas we are studying are often referred to as *system dynamics*.

Distinguishing between stocks and flows is sometimes difficult, and we will provide numerous examples below. As a starting point, you can think of stocks as representing physical entities which can accumulate and move around. However, in this age of computers, what used to be concrete physical entities have often become abstract. For example, money is often an important stock in many business processes. However, money is more often than not entries in a computer system, rather than physical dollar bills. In the pre-computer days, refunds in a department store might require the transfer of currency through a pneumatic tube; now they probably mean a computer credit to a MasterCard account. Nonetheless, the money is still a stock, and the transfer operation for the money is a flow.

Another way to distinguish stocks and flows is to ask what would happen if we could freeze time and observe the process. If we would still see a nonzero value for a quantity, then that quantity is a stock, but if the quantity could not be measured, then it is a flow. (That is, flows only occur over a period of time, and, at any particular instant, nothing moves.) For readers with an engineering systems analysis background, we use the term stock for what is called a *state variable* is engineering systems analysis.

Types of Stocks and Flows

Most business activities include one or more of the following five types of stocks: materials, personnel, capital equipment, orders, and money. The most visible signs of the operation of a process are often movements of these five types of stocks, and these are defined as follows:

Materials. This includes all stocks and flows of physical goods which are part of a production and distribution process, whether raw materials, in-process inventories, or finished products.

Personnel. This generally refers to actual people, as opposed, for example, to hours of labor.

Capital equipment. This includes such things as factory space, tools, and other equipment necessary for the production of goods and provision of services.

Orders. This includes such things as orders for goods, requisitions for new employees, and contracts for new space or capital equipment. Orders are typically the result of some management decision which has been made, but not yet converted into the desired result.

Money. This is used in the cash sense. That is, a flow of money is the actual transmittal of payments between different stocks of money.

The first three items above (materials, personnel, and capital equipment) are conceptually relatively straightforward because there is usually a physical entity corresponding to these. The last two items above (orders and money) are somewhat more subtle in this age of computers. Whether something is really money or just information about a monetary entry somewhere in a computer database may not be immediately obvious.

2.4 Information

The last element in the Figure 2.1b stock and flow diagram is the information link shown by the curved arrow from Potential Customers to sales. This arrow means that in some way information about the value of Potential Customers influences the value of sales. Furthermore, and equally important, the fact that there is no information arrow from Actual Customers to sales means that information about the value of Actual Customers does not influence the value of sales.

The creation, control, and distribution of information is a central activity of business management. The heart of the ongoing changes in business management is in changing the way that information is used. Perhaps nowhere is the impact of the computer on management potentially more significant. In a traditional hierarchical business organization, it can be argued that the primary role of much of middle management is to pass information up the hierarchy and orders down. This structure was required in pre-computer days by the magnitude of the communications problem in a large organization. With the current widespread availability of inexpensive computer-based analysis and communications systems, this large, expensive, and slow system for transmitting information is no longer adequate to retain competitive advantage. Business organizations are substantially changing the way they handle information, and thus the set of information links is a central component in most models of business processes oriented toward improving these processes.

The information links in a business process can be difficult to adequately model because of the abstract nature of these links. Materials, personnel, capital equipment, orders, and money usually have a physical representation. Furthermore, these quantities are conserved, and thus they can only flow to one place at a time. Information, on the other hand, can simultaneously flow to many places, and, particularly in computer-intensive environments, it can do this rapidly and with considerable distortion.

Practical experience is showing that modifying the information links in a business process can have profound impacts on the performance of the process. Furthermore, these impacts are often non-intuitive and can be dangerous. Some companies have discovered, for example, that computer-based information systems have not only not improved their performance, but in fact have degraded it. Doing large-scale experimentation by making *ad hoc* changes to a crucial aspect of an organization like the information links can be dangerous. The tools we discuss below provide a way to investigate the implications of such changes before they are implemented.

No one today would construct and fly an airplane without first carefully analyzing its potential performance with computer-based models. However, we routinely make major changes to our business organizations without such prior modeling. We seem to think that we can intuitively predict the performance of a changed organization, even though this organization is likely to be much more complex than an airplane. No one would take a ride on an airplane whose characteristics under all sorts of extreme conditions had not previously been analyzed carefully. Yet we routinely make significant changes to the structure of a business process and then take a ride in the resulting organization without this testing. The methods presented below aid in doing some testing before implementing changes to business processes.

2.5 Reference

J. W. Forrester, *Industrial Dynamics*, The MIT Press, Cambridge, Massachusetts, 1961.

CHAPTER 3

Simulation of Business Processes

The stock and flow diagram which have been reviewed in the preceding two chapters show more about the process structure than the causal loop diagrams studied in Chapter 1. However, stock and flow diagrams still don t answer some important questions for the performance of the processes. For example, the stock and flow diagram in Figure 2.1b shows more about the process structure than the causal loop diagram in Figure 2.1a, but it still doesn t answer some important questions. For example, how will the number of Potential Customers vary with time? To answer questions of this type, we must move beyond a graphical representation to consider the *quantitative* features of the process. In this example, these features include such things as the initial number of Potential and Actual Customers, and the specific way in which the sales flow depends on Potential Customers.

When deciding how to quantitatively model a business process, it is necessary to consider a variety of issues. Two key issues are how much detail to include, and how to handle uncertainties. Our orientation in these notes is to provide tools that you can use to develop better insight about key business processes. We are particularly focusing on the intermediate level of management decision making in an organization: Not so low that we must worry about things like specific placement of equipment in an manufacturing facility, and not so high that we need to consider decisions that individually put the company at risk.

This intermediate level of decision is where much of management s efforts are focused, and improvements at this level can significantly impact a company s relative competitive position. Increasingly, these decisions require a cross-functional perspective. Examples include such things as the impact on sales for a new product of capacity expansion decisions, relationships between financing and production capacity decisions, and the relationship between personnel policies and quality of service. Quantitatively considering this type of management decision may not require an extremely detailed model for business processes. For example, if you are considering the relationship between personnel policies and quality of service, it is probably not necessary to consider individual workers with their pay rates and vacation schedules. A more aggregated approach will usually be sufficient.

Furthermore, our primary interest is improving existing processes which are typically being managed intuitively, and to make these improvements in a reasonable amount of time with realistic data requirements. There is a long history of efforts to build highly detailed models, only to find that either the data required are not available, or the problem addressed has long since been solved by other means before the model was completed. Thus, we seek a relatively simple, straightforward quantitative modeling approach which can yield useful results in a timely manner.

The approach we take, which is generally associated with the field of system dynamics (Morecroft and Sterman 1994), makes two simplifying assumptions: 1) flows within processes are continuous, and 2) flows do not have a random component. By continuous flows, we mean that the quantity which is flowing can be infinitely finely divided, both with respect to the quantity of material flowing and the time period over which it flows. By not having a random component, we mean that a flow will be exactly specified if the values of the variables at the other end of information arrows into the flow are known. (A variable that does not have a random component is referred to as a *deterministic* variable.)

Clearly, the continuous flow assumption is not exactly correct for many business processes: You can t divide workers into parts, and you also can t divide new machines into parts. However, if we are dealing with a process involving a significant number of either workers or machines, this assumption will yield fairly accurate results and it substantially simplifies the model development and solution. Furthermore, experience shows that even when quantities being considered are small, treating them as continuous is often adequate for practical analysis.

The assumptions of no random component for flows is perhaps even less true in many realistic business settings. But, paradoxically, this is the reason that it can often be made in an analysis of business processes. Because uncertainty is so widely present in business processes, many realistic processes have evolved to be relatively insensitive to the uncertainties. Because of this, the uncertainty can have a relatively limited impact on the process. Furthermore, we will want any modifications we make to a process to leave us with something that continues to be relatively immune to randomness. Hence, it makes sense in many analyses to assume there is no uncertainty, and then test the consequences of possible uncertainties.

Practical experience indicates that with these two assumptions, we can substantially increase the speed with which models of business processes can be built, while still constructing models which are useful for business decision making.

3.1 Equations for Stocks

With the continuous and deterministic flow assumptions, a business process is basically modeled as a plumbing system. You can think of the stocks as tanks full of a liquid, and the flows as valves, or, perhaps more accurately, as pumps that control the rate of flow between the tanks. Then, to completely specify the equations for a process model you need to give 1) the initial values of each stock, and 2) the equations for each flow.

We will now apply this approach to the advertising stock and flow diagram in Figure 2.1b. To do this, we will use some elementary calculus notation. However, fear not! You do not have to be able to carry out calculus operations to use this approach. Computer methods are available to do the required operations, as is discussed below, and the calculus discussion is presented for those who wish to gain a better understanding of the theory behind the computer methods.

The number of Potential Customers at any time t is equal to the number of Potential Customers at the starting time minus the number that have flowed out due to sales. If sales is measured in customers per unit time, and there were initially 1,000,000 Potential Customers, then

$$\text{Potential Customers}(t) = 1\ 000\ 000 - \int_0^t \text{sales}(\tau)\, d\tau \qquad (3\ 1)$$

where we assume that the initial time is $t = 0$, and τ is the dummy variable of integration. Similarly, if we assume that there were initially zero Actual Customers, then

$$\text{Actual Customers} = \int_0^t \text{sales}(\tau)\, d\tau \qquad (3\ 2)$$

The process illustrated by these two equations generalizes to any stock: The stock at time t is equal to the initial value of the stock at time $t = 0$ plus the integral of the flows into the stock minus the flows out of the stock. Notice that once we have drawn a stock and flow diagram like the one shown in Figure 2.1b, then a clever computer program could enter the equation for the value of any stock at any time without you having to give any additional information except the initial value for the stock. In fact, system dynamics simulation packages automatically enter these equations.

3.2 Equations for Flows

However, you must enter the equation for the flows yourself. There are many possible flow equations which are consistent with the stock and flow diagram in Figure 2.1b. For example, the sales might be equal to 25,000 customers per month until the number of Potential Customers drops to zero. In symbols,

$$\text{sales}(t) = \begin{cases} 25\ 000 & \text{Potential Customers}(t) > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (3\ 3)$$

A more realistic model might say that if we sell a product by advertising to Potential Customers, then it seems likely that some specified percentage of Potential Customers will buy the product during each time unit. If 2.5 percent of the Potential Customers make a purchase during each month, then the equation for sales is

$$\text{sales}(t) = 0\ 025 \times \text{Potential Customers}(t) \qquad (3\ 4)$$

(Notice that with this equation the initial value for sales will be equal to 25,000 customers per month.)

3.3 Solving the Equations

If you are familiar with solving differential equations, then you can solve equations 3.1 and 3.2 in combination with either equation 3.3 or equation 3.4 to obtain a graph of Potential Customers over time. However, it quickly becomes infeasible to solve such equations by hand as the number of stocks and flows increases, or if the equations for the stocks are more complex than those shown in equation 3.3 and 3.4. Thus, computer solution methods are almost always used.

We will illustrate how this is done using the Vensim simulation package. With this package, as with most PC-based system dynamics simulation systems, you typically start by entering a stock and flow diagram for the model. In fact, the stock and flow diagram shown in Figure 2.1b was created using Vensim. You then enter the initial values for the various stocks into the model, and also the equations for the flows. Once this is done, you then tell the system to solve the set of equations. This solution process is referred to as *simulation*, and the result is a time-history for each of the variables in the model. The time history for any particular variable can be displayed in either graphical or tabular form.

Figure 3.1 shows the Vensim equations for the model using equations 3.1 and 3.2 for the two stocks, and either equation 3.3 (in Figure 3.1a) or equation 3.4 (in Figure 3.1b) for the flow sales. These equations are numbered and listed in alphabetical order.

Note that equation 1 in either Figure 3.1a or b corresponds to equation 3.2 above, and equation 4 is either Figure 3.1a or b corresponds to equation 3.1 above. These are the equations for the two stock variables in the model. The notation for these is straightforward. The function name `INTEG` stands for integration, and it has two arguments. The first argument includes the flows into the stock, where flows out are entered with a minus sign. The second argument gives the initial value of the stock.

Equation 5 in Figure 3.1a corresponds to equation 3.3 above, and equation 5 in Figure 3.1b corresponds to equation 3.4 above. These equations are for the flow variable in the model, and each is a straightforward translation of the corresponding mathematical equation.

Equation 3 in either Figure 3.1a or b sets the lower limit for the integrals. Thus, the equation `INITIAL TIME = 0` corresponds to the lower limits of $t = 0$ in equations 3.1 and 3.2 above. Equation 2 in either Figure 3.1a or b sets the last time for which the simulation is to be run. Thus, with `FINAL TIME = 100`, the values of the various variables will be calculated from the `INITIAL TIME` (which is zero) until a time of 100 (that is, $t = 100$).

Equations 6 and 7 in either Figure 3.1a or b set characteristics of the simulation process.

```
(1)  Actual Customers = INTEG( sales , 0)
(2)  FINAL TIME = 100
(3)  INITIAL TIME = 0
(4)  Potential Customers = INTEG( - sales , 1e+006)
(5)  sales = IF THEN ELSE ( Potential Customers > 0, 25000, 0)
(6)  SAVEPER = TIME STEP
(7)  TIME STEP = 1
```

a. Equations with constant sales

```
(1)  Actual Customers = INTEG( sales , 0)
(2)  FINAL TIME = 100
(3)  INITIAL TIME = 0
(4)  Potential Customers = INTEG( - sales , 1e+006)
(5)  sales = 0.025 * Potential Customers
(6)  SAVEPER = TIME STEP
(7)  TIME STEP = 1
```

b. Equations with proportional sales

Figure 3.1 *Vensim equations for advertising model*

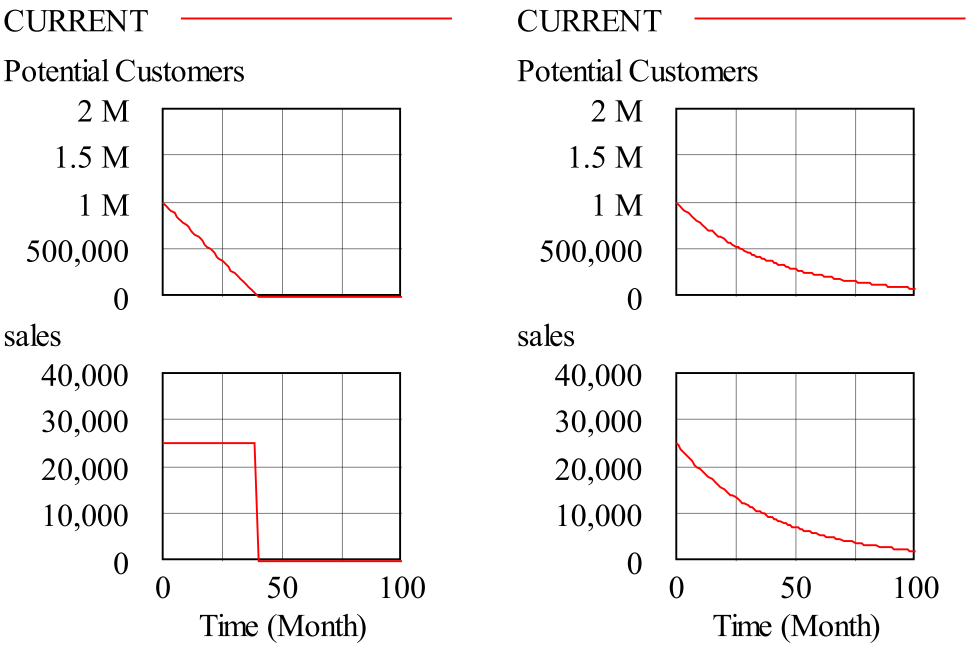
3.4 Solving the Model

The time histories for the sales and Potential Customers variables are shown in Figure 3.2. The graphs in Figure 3.2a were produced using the equations in Figure 3.1a, and the graphs in Figure 3.2b were produced using the equations in Figure 3.1b. We see from Figure 3.2a that sales stay at 25,000 customers per month until all the Potential Customers run out at time $t = 40$. Then sales drop to zero. Potential Customers decreases linearly from the initial one million level to zero at time $t = 40$. Although not shown in this figure, it is easy to see that Actual Customers must increase linearly from zero initially to one million at time $t = 40$.

In Figure 3.2b, sales decreases in what appears to be an exponential manner from an initial value of 25,000, and similarly Potential Customers also decreases in an exponential manner. (In fact, it can be shown that these two curves are exactly exponentials.)

3.5 Some Additional Comments on Notation

In the Figure 2.1b stock and flow diagram, the two stock variables Potential Customers and Actual Customers are written with initial capital letters on each word. This is recommended practice, and it will be followed below. Similarly, the flow sales is written in all lower case, and this is recommended practice.



a. Constant sales

b. Proportional sales

Figure 3.2 *Time histories of sales and Potential Customers*

While stock and flow variables are all that are needed in concept to create any stock and flow diagram, it is often useful to introduce additional variables to clarify the process model. For example, in a stock and flow diagram for the Figure 3.1b model, it might make sense to introduce a separate variable name for the sales fraction (which was given as 0.025 in equation 3.4. This could clarify the structure of the model, and also expedites sensitivity analysis in most of the system dynamics simulation packages.

Such additional variables are called *auxiliary variables*, and examples will be shown below. It is recommended that an auxiliary variable be entered in ALL CAPITAL LETTERS if it is a constant. Otherwise, it should be entered in all lower case letters just like a flow variable, except in one special case. This is the case where the variable is not a constant but is a prespecified function of time (for example, a sine function). In this case, the variable name should be entered with the FIRst three letters capitalized, and the remaining letters in lower case.

This notation allows you to quickly determine important characteristics of variable in a stock and flow diagram from the diagram without having to look at the equations that go with the diagram.

3.6 Reference

J. D. W. Morecroft and J. D. Sterman, editors, *Modeling for Learning Organizations*, Productivity Press, Portland, OR, 1994.

CHAPTER 4

Basic Feedback Structures

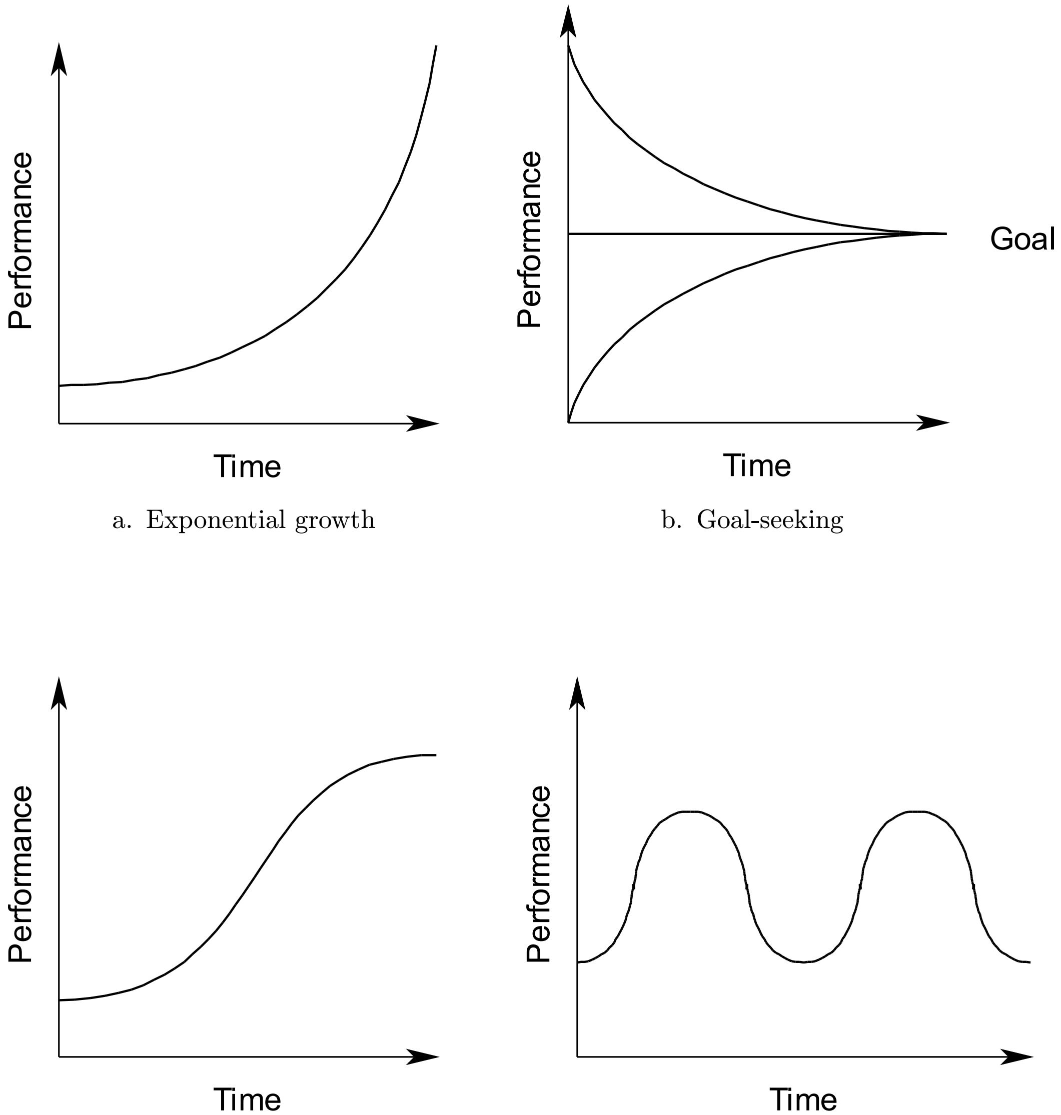
This chapter reviews some common patterns of behavior for business processes, and presents process structures which can generate these patterns of behavior. Many interesting patterns of behavior are caused, at least in part, by *feedback*, which is the phenomenon where changes in the value of a variable indirectly influence future values of that same variable. *Causal loop diagrams* (Richardson and Pugh 1981, Senge 1990) are a way of graphically representing feedback structures in a business process with which some readers may be familiar. However, causal loop diagrams only suggest the possible modes of behavior for a process. By developing a stock and flow diagram and corresponding model equations, it is possible to estimate the actual behavior for the process.

Figure 4.1 illustrates four patterns of behavior for process variables. These are often seen individually or in combination in a process, and therefore it is useful to understand the types of process structures that typically lead to each pattern.

4.1 Exponential Growth

Exponential growth, as illustrated in Figure 4.1a, is a common pattern of behavior where some quantity feeds on itself to generate ever increasing growth. Figure 4.2 shows a typical example of this the growth of savings with compounding interest. In this case, increasing interest earnings lead to an increase in Savings, which in turn leads to greater interest because interest earnings are proportional to the level of Savings, as shown in equation 3 of Figure 4.2b. Figure 4.2c shows the characteristic upward-curving graph that is associated with this process structure. This is referred to as an exponential curve because it can be demonstrated that it follows the equation of the exponential function. (Remember that the cloud at the left side of Figure 4.2a means that we are not explicitly modeling the source of the interest.)

While it is possible to use standard calculus methods to solve for the variables in this model, we will not do this because this structure is typically only one



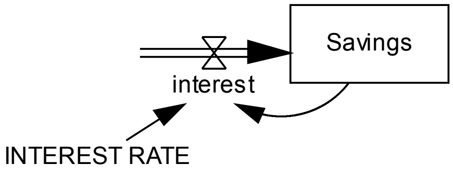
a. Exponential growth

b. Goal-seeking

c. S-shaped

d. Oscillation

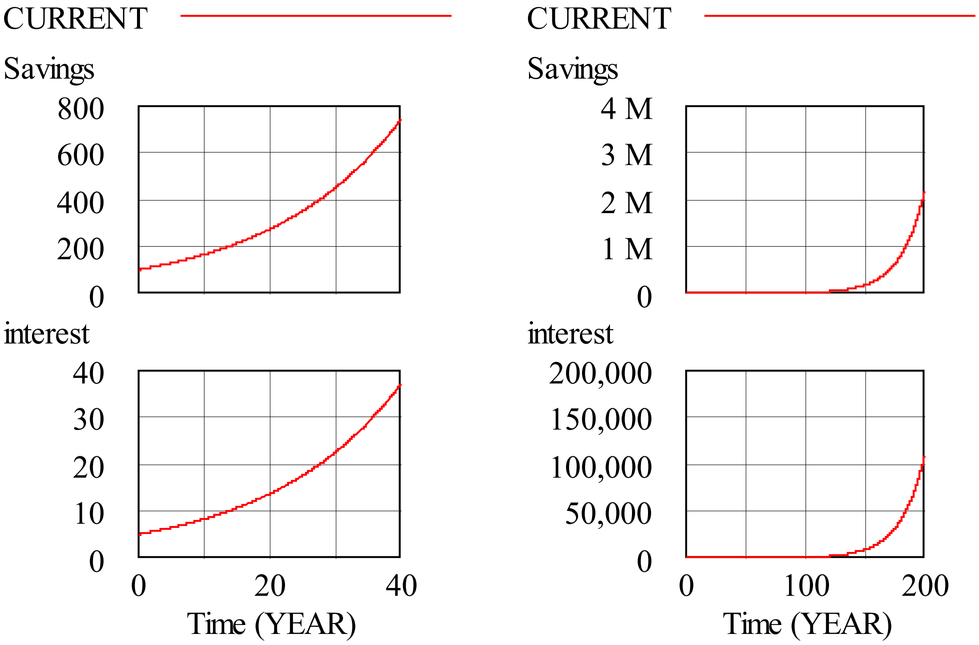
Figure 4.1 *Characteristic patterns of system behavior*



a. Stock and flow diagram

```
(1)  FINAL TIME  = 40
(2)  INITIAL TIME  = 0
(3)  interest = INTEREST RATE*Savings
(4)  INTEREST RATE = 0.05
(5)  SAVEPER  = TIME STEP
(6)  Savings = INTEG(interest,100)
(7)  TIME STEP  = 0.0625
```

b. Vensim equations



c. Forty year horizon

d. Two hundred year horizon

Figure 4.2 *Exponential growth feedback process*

component of a more complex process in realistic settings. The models of those more complex processes usually cannot be solved in closed form, and therefore we have shown the Vensim simulation equations used to simulate this model in Figure 4.2b.

Figure 4.2d shows another characteristic of exponential growth processes. In this diagram, the time period considered has been extended to 200 years. When this is done, we see that exponential growth over an extended period of time displays a phenomenon where there appears to be almost no growth for a period, and then the growth explodes. This happens because with exponential growth the period which it takes to double the value of the growing variable (called the doubling time) is a constant regardless of the current level of the variable. Thus, it will take just as long for the variable to double from 1 to 2 as it does to double from 1,000 to 2,000, or from 1,000,000 to 2,000,000. Hence, while the variables in Figure 4.2d are growing at a steady exponential rate during the entire 200 year period, because of the large vertical scale necessary for the graph in order to show the values at the end of the period, it is not possible to see the growth during the early part of the period.

4.2 Goal Seeking

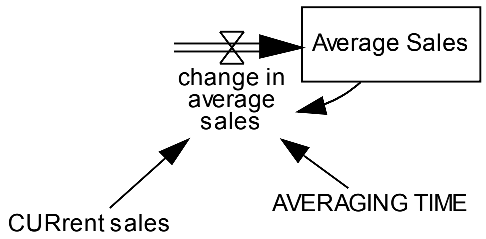
Figure 4.1b displays goal seeking behavior in which a process variable is driven to a particular value. Figure 4.3 presents a process which displays this behavior. As CURrent sales change, the level of Average Sales moves to become the same as CURrent sales. However, it moves smoothly from its old value to the CURrent sales value, and this is the origin of the name Average Sales for this variable. (In fact, this structure can be used to implement the SMOOTH function which we have previously seen.)

Figure 4.3c and Figure 4.3d show what happens when CURrent sales takes a step up (in Figure 4.3c) or a step down (in Figure 4.3d). While it is somewhat hard to see in these graphs, CURrent sales is plotted with a solid line which jumps at time 10. Until that time, Average Sales have been the same as CURrent sales, then they diverge since it takes a while for Average Sales to smoothly move to again become the same as CURrent sales.

Equation 3 in Figure 4.3b shows the process which drives Average Sales toward the value of CURrent sales. If Average Sales are below CURrent sales, then there is flow into the Average Sales stock, while if Average Sales are above CURrent sales, then there is flow out of the Average Sales stock. In either case, the flow continues as long as Average Sales differs from CURrent sales.

The rate at which the flow occurs depends on the constant AVERAGING TIME. The larger the value of this constant, the slower the flow into or out of Average Sales, and hence the longer it takes to bring the value of Average Sales to that of CURrent sales.

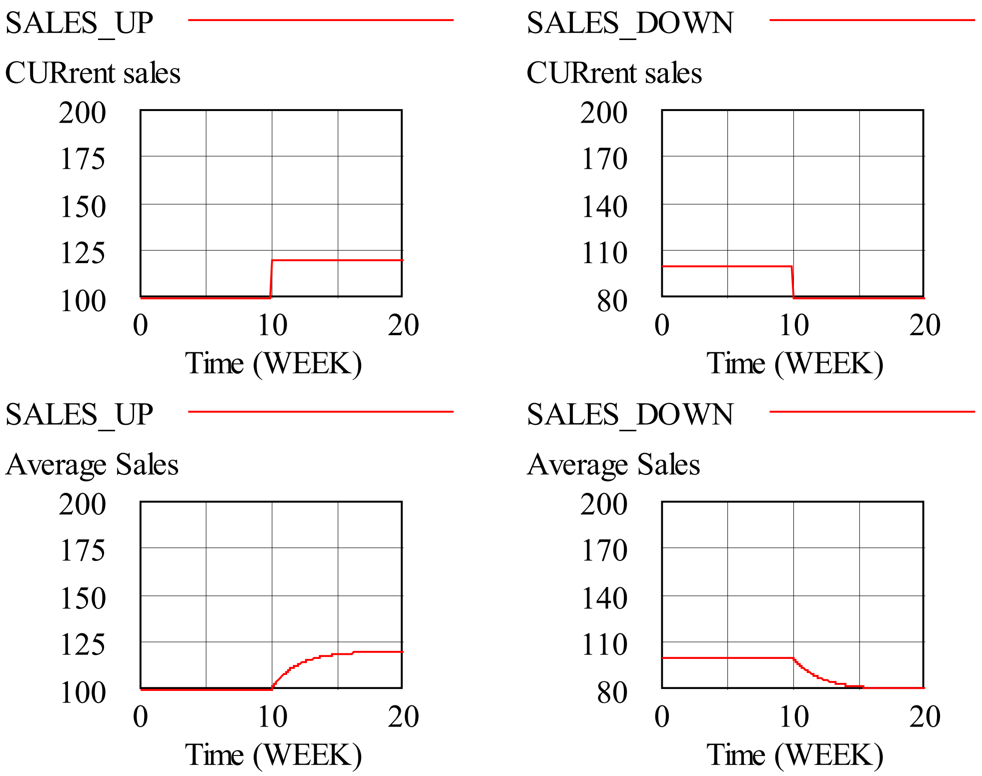
It is possible to solve the equations for a goal seeking process to show that the equation for the curve of the variable moving toward a goal (Average Sales in Figure 4.3) has an exponential shape. However, as with exponential growth, a goal seeking process is often only a part of a larger process for which it is not



a. Stock and flow diagram

```
(1)  Average Sales = INTEG(change in average sales,100)
(2)  AVERAGING TIME = 2
(3)  change in average sales = (CURrent sales-Average Sales)
     /AVERAGING TIME
(4)  CURrent sales = 100+STEP(20,10)
(5)  FINAL TIME  = 20
(6)  INITIAL TIME  = 0
(7)  SAVEPER  = TIME STEP
(8)  TIME STEP  = 0.0625
```

b. Vensim equations



c. Sales move up

d. Sales move down

Figure 4.3 *Goal seeking process*

possible to obtain a simple solution, and thus we show the simulation equations for this process.

Note that the process shown in Figure 4.4 is a negative feedback process. As the value of change in average sales increases, this causes an increase in the value of Average Sales, which in turn leads to a decrease in the value of change in average sales.

4.3 S-shaped Growth

Exponential growth can be exhilarating if it is occurring for something that you makes you money. The future prospects can seem endlessly bright, with things just getting better and better at an ever increasing rate. However, there are usually limits to this growth lurking somewhere in the background, and when these take effect the exponential growth turns into goal seeking behavior, as shown in Figure 4.1c.

Figure 4.4 shows a business process structure which can lead to this s-shaped growth pattern. This illustrates a possible structure for the sale of some sort of durable good for which word of mouth from current users is the source of new sales. This might be called a contagion model of sales being a user of the product is contagious to other people! We assume that there is a specified INITIAL TOTAL RELEVANT POPULATION of potential customers for the product. (This is the limit that will ultimately stop growth in Actual Customers.) At any point in time, there is a total of Potential Customers of potential users who have not yet bought the product.

Visualize the process of someone in the Potential Customers group being converted into an Actual Customer as follows: The two groups of people who are in the Actual Customers group and in the Potential Customers group circulate among the larger general population and from time to time they make contact. When they make contact, there is some chance that the comments of the person who is an Actual Customer will cause the person who is in the Potential Customers to buy the product.

The model shown in Figure 4.4 assumes that for each such contact between a person in the Actual Customers and a person in the susceptible population there will be a number of sales equal to SALES PER CONTACT, which will probably be less than one in most realistic settings. The number of sales per unit of time will be equal to SALES PER CONTACT times the number of contacts per unit of time between persons in the Actual Customers and Potential Customers groups. But with the assumed random contacts between persons in the two groups, the number of contacts per unit time will be proportional to both the size of the Actual Customers group and the size of the Potential Customers group. Hence sales is proportional to the product of Actual Customers and Potential Customers. The proportionality constant is called BASE CONTACT RATE in Figure 4.4, and it represents the number of contacts per unit time when each of the two groups has a size equal to one. (That is, it is the number of contacts per unit time between any specified member of the Actual Customers group and any specified member of the Potential Customers group.)

The argument in the last paragraph for a multiplicative form for the sales equation (as shown in equation 6 of Figure 4.4b) was somewhat informal. A more formal argument can be made by using probability theory. Select a short enough period of time so that at most one contact can occur between any persons in the Actual Customers group and the Potential Customers group regardless of how large these groups are. Then assume that the probability that any specified member of the Actual Customers group will contact any specified member of the Potential Customers group during this period is some (unspecified) number p. Then, if this probability is small enough (which we can make it by reducing the length of the time period considered), the probability that the specified member of the Actual Customers group will contact *any* member of the susceptible population is equal to $p \times$ Potential Customers.

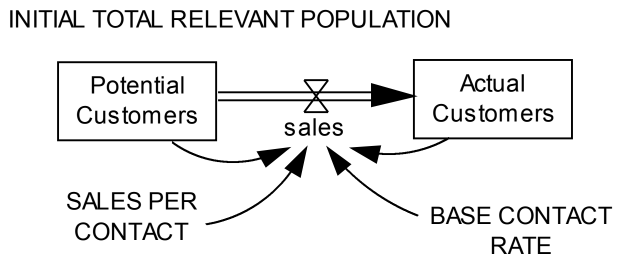
Assuming that this probability is small enough for any individual member of the Actual Customers population, then the probability that *any* member of the Actual Customers population will contact a member of the Potential Customers population is just this probability times the number of members in the Actual Customers population, or

$$p \times \text{susceptible population} \times \text{Actual Customers}$$

Assuming the interaction process between the two groups is a Poisson process and the probability of a successful interaction (that is, a sale) is fixed, then the sale process is a random erasure process on a Poisson process and hence is also a Poisson process. Thus, the expected number of sales per unit time is proportional to the probability expression above, and hence to the product of Actual Customers and susceptible population. This is the form assumed in equation 6 of Figure 4.4b.

Figure 4.4c shows the resulting pattern for the number of Actual Customer, as well as the sales. This s-shaped pattern is seen with many new products. First the process grows exponentially, and then it levels off. Sales also grow exponentially for a while, and then they decline. This can be a difficult process to manage because the limit to growth is often not obvious while the exponential growth is under way. For example, when a new consumer product like the compact disk player is introduced, what is the INITIAL TOTAL RELEVANT POPULATION of possible customers for the product? The difference between a smash hit like the compact disk player and a dud like quadraphonic high fidelity sound systems can be hard to predict.

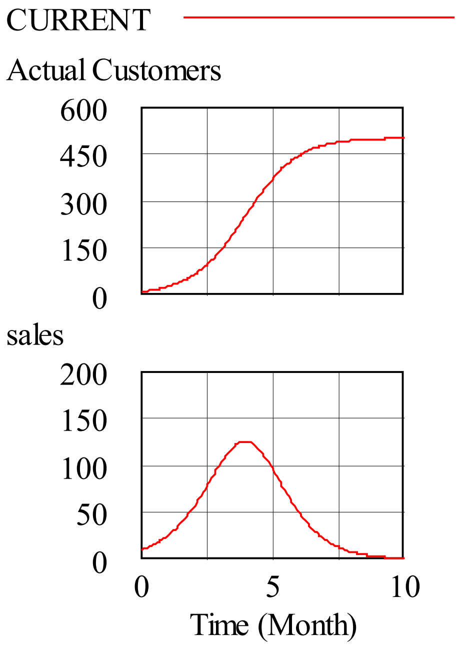
Note that there are two feedback loops, one positive and one negative, that involve the variable sales in the Figure 4.5a diagram. The positive loop involves sales and Actual Customers. The negative loop involves sales and Potential Customers. At first the positive loop dominates, but later the negative loop comes to dominate. (There is another feedback loop through the initial condition on Potential Customers, which depends on Actual Customers. However, this is not active once the process starts running.)



a. Stock and flow diagram

```
(01)  Actual Customers = INTEG(sales, 10)
(02)  BASE CONTACT RATE = 0.02
(03)  FINAL TIME  = 10
(04)  INITIAL TIME  = 0
(05)  Potential Customers = INTEG(-sales,
        INITIAL TOTAL RELEVANT POPULATION - Actual Customers)
(06)  sales = BASE CONTACT RATE * SALES PER CONTACT
              * Actual Customers * Potential Customers
(07)  SALES PER CONTACT = 0.1
(08)  SAVEPER  = TIME STEP
(09)  TIME STEP  = 0.0625
(10)  INITIAL TOTAL RELEVANT POPULATION = 500
```

b. Vensim equations



c. Customer and sales performance

Figure 4.4 *S-shaped growth process*

4.4 S-shaped Growth Followed by Decline

Figure 4.5 shows a process model for a variation on s-shaped growth where the leveling off process is followed by decline. In this process, it is assumed that some Actual Customers and some Potential Customers permanently quit. Such a process might make sense for a new fad durable good which comes on the market. In such a situation, there may be a large INITIAL TOTAL RELEVANT POPULATION of possible customers but some of those who purchase the product and become Actual Customers may lose interest in the product and cease to discuss it with Potential Customers. Similarly, some Potential Customers lose interest before they are contacted by Actual Customers. Gradually both sales and use of the product will decline.

In equation 5 of Figure 4.5, the quitting processes for both Potential Customers and Actual Customers are shown as exponential growth processes running in reverse. That is, the number of Actual Customers *leaving* is proportional to the number of Actual Customers rather than the number *arriving*, as in a standard exponential growth process. Similarly, the number of Potential Customers leaving is proportional to the number of Potential Customers. This type of departure process also can be viewed as a balancing process with a goal of zero, and it is sometimes called *exponential decline* or *exponential decay.* From Figure 4.5c, we see that this exponential decline process eventually leads to a decline in the number of Actual Customers.

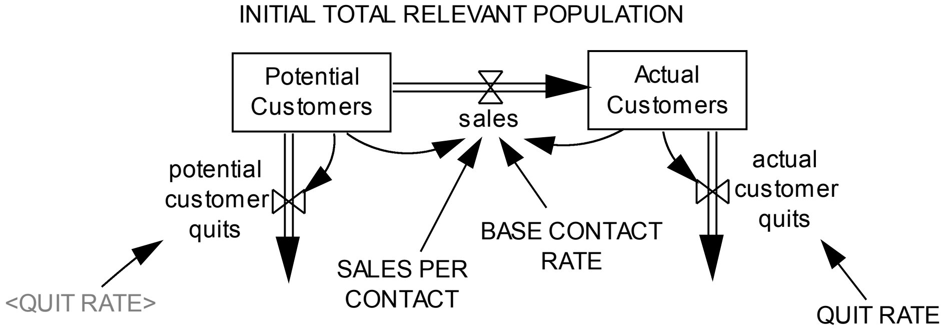
4.5 Oscillating Process

The Figure 4.6a stock and flow diagram is a simplified version of a production-distribution process. In this process, the retailer orders to the factory depend on both the retail sales and the Retail Inventory level. The factory production process is shown as immediately producing to fulfill the retailer orders, but there is a delay in the retailer receiving the product because of shipping delays.

In this process, RETail sales are 100 units per week until week 5, at which point they jump to 120 units and remain there for the rest of the simulation run. We see from Figure 4.6c that there are substantial oscillations in key variables of the process.

Unless there are very unusual flow equations, there must be at least two stocks in a process for the process to oscillate. Furthermore, the degree of oscillation is usually impacted by the delays in the process. The important role of stocks and delays in causing oscillation is one of the factors behind moves to just in time production systems and computer-based ordering processes. These approaches can reduce the stocks in a process and also can reduce delays.

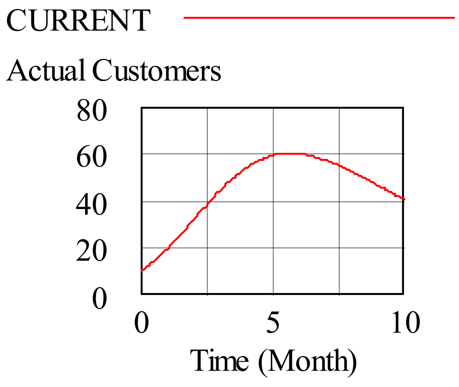
Figure 4.7 illustrates another aspect of oscillating systems. The process in Figure 4.7 is identical to that in Figure 4.6 except that the RETail sales function has been changed from a step to a sinusoid. Thus, sales are stable at 100 units per week until week 5, and then sales vary sinusoidally with an amplitude above and below 100 units per week of 20. The results for three different cycle lengths are



a. Stock and flow diagram

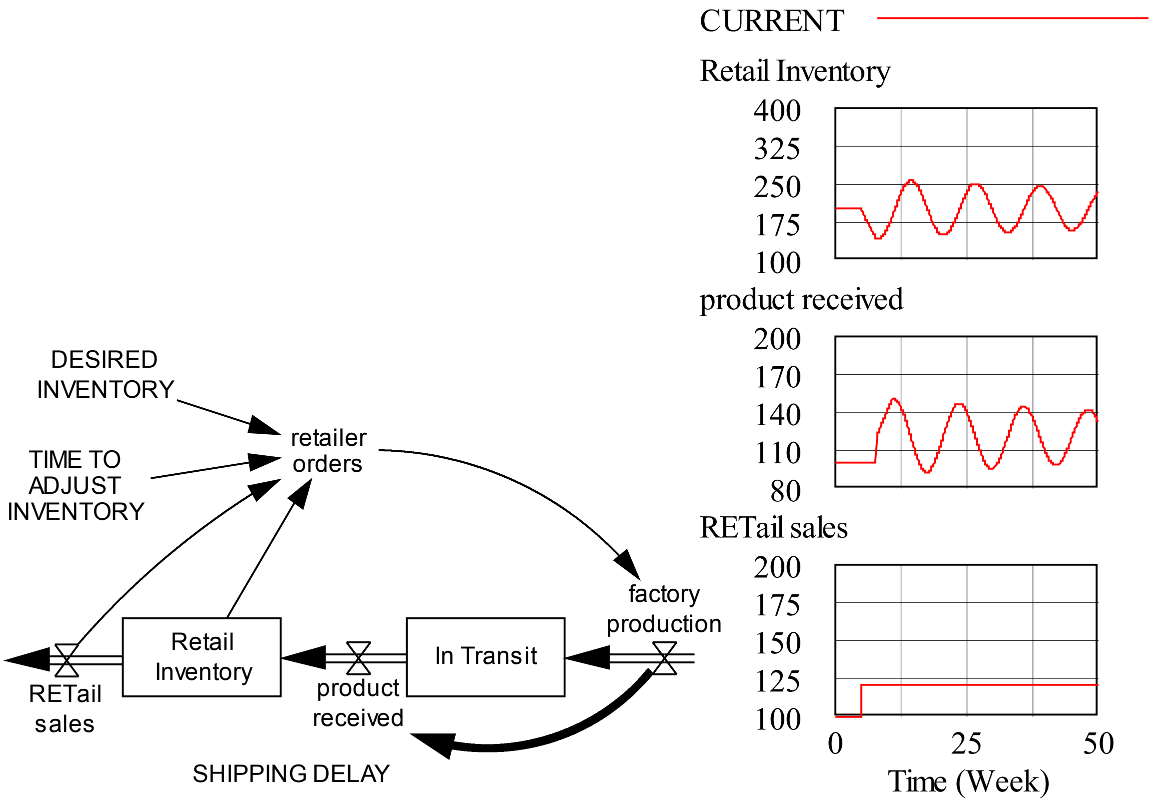
```
(01)  actual customer quits = QUIT RATE * Actual Customers
(02)  Actual Customers = INTEG(sales - actual customer quits, 10)
(03)  BASE CONTACT RATE = 0.02
(04)  FINAL TIME  = 10
(05)  INITIAL TIME  = 0
(06)  INITIAL TOTAL RELEVANT POPULATION = 500
(07)  potential customer quits = QUIT RATE * Potential Customers
(08)  Potential Customers = INTEG(-sales - potential customer quits,
        INITIAL TOTAL RELEVANT POPULATION - Actual Customers)
(09)  QUIT RATE = 0.2
(10)  sales = BASE CONTACT RATE * SALES PER CONTACT
                * Actual Customers * Potential Customers
(11)  SALES PER CONTACT = 0.1
(12)  SAVEPER  = TIME STEP
(13)  TIME STEP  = 0.0625
```

b. Vensim equations



c. Customer performance

Figure 4.5 *S-shaped growth followed by decline*



a. Stock and flow diagram

c. Process oscillations

```
(01)  DESIRED INVENTORY = 200
(02)  factory production = retailer orders
(03)  FINAL TIME  = 50
(04)  In Transit = INTEG(factory production-orders received, 300)
(05)  INITIAL TIME  = 0
(06)  product received = DELAY FIXED(factory production,
                         SHIPPING DELAY, factory production)
(07)  Retail Inventory = INTEG(product received-retail sales, 200)
(08)  RETail sales = 100 + STEP(20, 5)
(09)  retailer orders = retail sales+ (DESIRED INVENTORY
                     - Retail Inventory) / TIME TO ADJUST INVENTORY
(10)  SAVEPER  = TIME STEP
(11)  SHIPPING DELAY = 3
(12)  TIME STEP  = 0.0625
(13)  TIME TO ADJUST INVENTORY = 2
```

b. Vensim equations

Figure 4.6 *Oscillating feedback process*

shown in Figure 4.7c. The RUN4 results are for a cycle length of 4 weeks (that is, a monthly cycle). The RUN13 results are for a 13 week (that is, quarterly) cycle, and the RUN52 results are for a 52 week (that is, annual) cycle.

Notice that the amplitude of the variations in Retail Inventory and product received are different for the three different cycle lengths. The amplitude is considerably greater for the 13 week cycle than for either the 4 week or 52 week cycles. This is true even though the amplitude of the RETail sales is the same for each cycle length.

Now go back and examine the curves in Figure 4.6c which shows the response of this process to a step change in retail sales. Note in particular that the cycle length for the oscillations is around 12 weeks. A cycle length at which a process oscillates in response to a step input is called a *resonance* of the process, and the inverse of the cycle length is called a *resonant frequency.* Thus, a resonant frequency for this process is 1 12 = 0 0833 cycles per week.

A process will generally respond with greater amplitude to inputs which vary with a frequency that is at or near a resonant frequency. Thus, it is to be expected that the response shown in Figure 4.7c for the sinusoidal with a 13 week cycle is greater than the responses for the sinusoids with 4 and 52 week cycles.

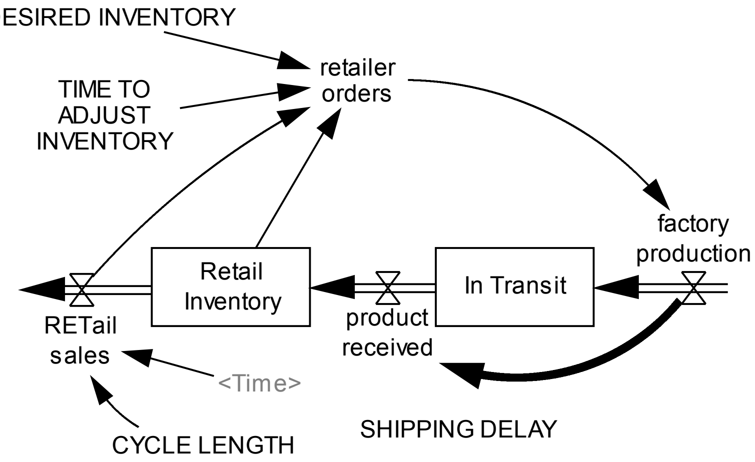
In engineered systems, an attempt is often made to keep the resonant frequencies considerably different from the usual variations that are found in operation. This is because of the large responses that such systems typically make to inputs near their resonant frequencies. This can be annoying, or even dangerous. (Have you ever noticed the short period of vibration that some planes go through just after takeoff? This is a resonance phenomena.)

Unfortunately, the resonant frequencies for many business processes are in the range of variations that are often found in practice. This has two undesirable aspects. First, it means that the amplitude of variations is greater than it might otherwise be. Second, it may lead managers to assume there are external causes for the variations. Suppose that in a particular process these oscillations have periods that are similar to some natural time period like a month, quarter, or year. In such a situation, it can be easy to assume that there is some external pattern that has such a period, and start to organize your process to such a cycle. This can make the oscillations worse. For example, consider the traditional yearly cycle in auto sales. Is that due to real variations in consumer demand, or is it created by the way that the auto companies manage their processes?

4.6 References

G. P. Richardson and A. L. Pugh III, *Introduction to System Dynamics Modeling with DYNAMO*, Productivity Press, Cambridge, Massachusetts, 1981.

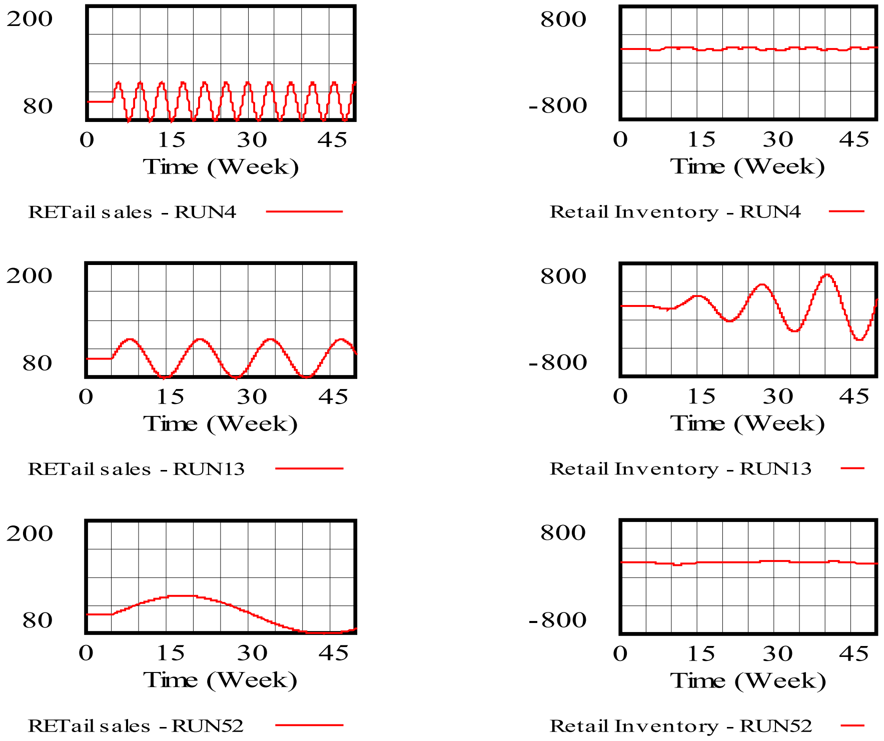
P. M. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday Currency, New York, 1990.



a. Diagram

```
CYCLE LENGTH = 13
RETail sales
  = 100 + STEP(20, 5) * SIN(2 * 3.14159 * (Time-5) / CYCLE LENGTH)
```

b. Changes to Figure 4.6 Vensim equations



c. Process oscillations (4, 13, and 52 week cycles)

Figure 4.7 *Performance with oscillating retail sales*

CHAPTER 5

Developing a Model

This chapter illustrates how a simulation model is developed for a business process. Specifically, we develop and investigate a model for a simple production-distribution system. Such systems are at the heart of most companies that make and sell products, and similar systems exist in most service-oriented businesses. Regardless of where you work within most companies, it is useful to understand the sometimes counterintuitive behavior that is possible in a production-distribution system. As we will see, difficulties in a production-distribution system that are often attributed to external events can be caused by the internal structure of the system.

The purpose of this example is to familiarize you with what is required to build a simulation model, and how such a model can be used. Some of the details presented below may not be totally clear at this point. In later chapters, we will investigate in further detail a number of topics that help clarify these details and assist you in building your own models.

A basic stock and flow diagram for the system we will consider is shown in Figure 5.1[1]. There are two flow processes: The production process shown at the top of the figure with a flow to the right, and the distribution system shown at the bottom of the figure with a flow to the left. The production system is a flow of *orders*, while the distribution system is a flow of *materials*. The two processes are tied together by *factory production*, as shown at the right side of the figure. As items are produced, the orders for these items are removed from the Factory Order Backlog, and the items are placed into Retail Inventory.

Note the use of the small clouds which are shown at the right and left ends of the production and distribution processes. These clouds represent either a *source* or a *sink* of flow which is outside the process that we are considering. For example, the cloud in the upper right corner of the figure shows that we are not considering in our analysis what happens to orders once they have initiated factory production. (In an actual system, the orders probably continue to flow into a billing process. That is outside the bounds of what we are interested in

[1] The models in this chapter are adapted from Jarmain (1963), pp. 118 124.

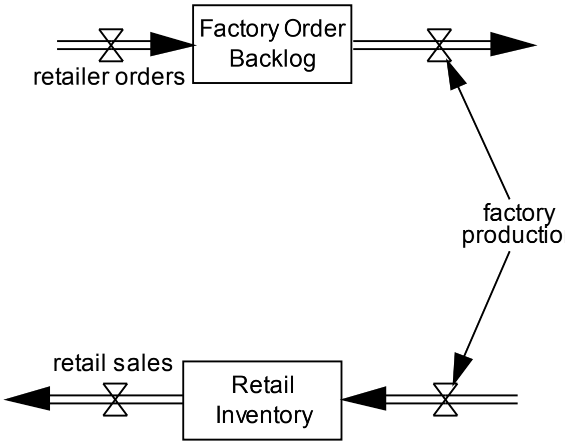


Figure 5.1 *A simple production-distribution system*

here, and therefore we simply show a cloud into which the orders disappear a sink .)

The production-distribution system shown in Figure 5.1 is simpler than most real systems. These often involve multiple production stages, and also multiple distribution stages (for example, distributor, wholesaler, and retailer), each of which has an inventory of goods. Thus, it might seem that this example is too simple to teach us much that is interesting about real-world production-distribution processes. Surely our intuition will be sufficient to quickly find a good way to run this system! Perhaps not. As we will see, this simplified production-distribution system is still complicated enough to produce counter-intuitive behavior. Furthermore, this behavior is typical of what is seen in real production-distribution systems.

We will study the policy that the retailer uses to place orders with the factory, and we will develop five different models to investigate different policies for placing these orders. As we will see, it is not necessarily straightforward to develop an ordering policy that has desirable characteristics.

5.1 The First Model

Figure 5.2, which has three parts, shows the first model for the production-distribution system, and the four other models which follow will also be shown with analogous three-part figures. Figure 5.2a shows the stock and flow diagram for the first model, Figure 5.2b shows the Vensim equations for this model, and Figure 5.2c shows the performance of key variables within the process.

Figure 5.2a was developed from the Figure 5.1 stock and flow diagram by adding several information flows. At the left-center side of the diagram, the auxiliary variable average retail sales has been added, along with an auxiliary constant TIME TO AVERAGE SALES. In the lower left side of the figure, another auxiliary variable TESt input has been added. Since this variable

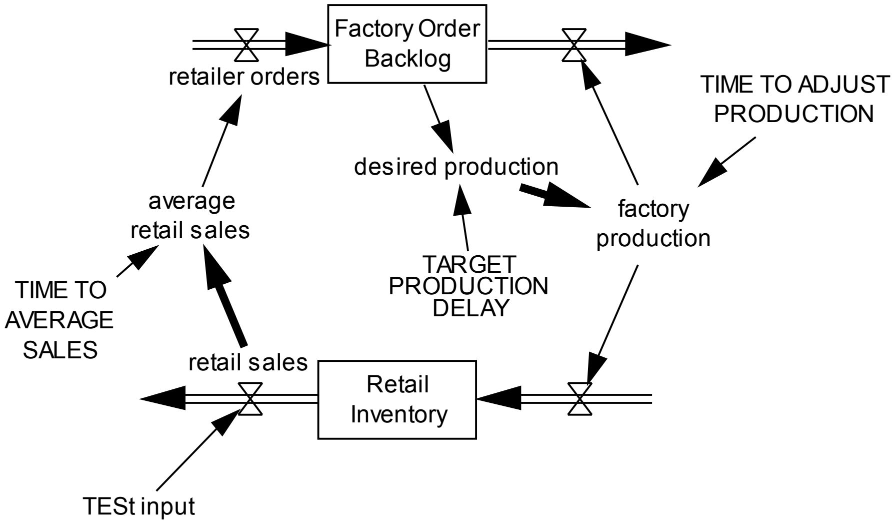


Figure 5.2a *Stock and flow diagram for first model*

name starts with three capital letters, we know that it varies over time in a prespecified manner. A variable which has a prespecified variation over time is called an *exogenous variable.*

In the center of the Figure 5.2a diagram, an auxiliary variable desired production has been added, along with an auxiliary constant TARGET PRODUCTION DELAY. Finally, at the right-center of the diagram, the auxiliary constant TIME TO ADJUST PRODUCTION has been added.

Factory Production

The Figure 5.2a diagram presents a particular procedure for how ordering is done by the retailer, as well as how production is managed. We are going to focus on different ordering policies for the retailer in our analysis, but first we will develop a model for how factory production is managed. This is shown by the variables at the right side of Figure 5.2a. From the information arrows shown there, we see that there is a desired production which depends on the Factory Order Backlog and the TARGET PRODUCTION DELAY. This desired production is then used to set the actual factory production, but there is some delay in adjusting factory production, as shown by the constant TIME TO ADJUST PRODUCTION. In this diagram, a delay in an information flow is indicated by using a thicker arrow. Such thicker arrows are shown pointing from retail sales to average retail sales, and from desired production to factory production.

In this simplified model for production, there is no inventory within the factory instead, the flow of the production system is adjusted to attempt to maintain a TARGET PRODUCTION DELAY, which is measured in weeks.

That is, the philosophy underlying this production system is that the retail orderer should be able to predict the length of time it will take to receive an order that is placed. Thus, if the TARGET PRODUCTION DELAY is two weeks, the factory will attempt to set production so that the current Factory Order Backlog will be cleared in two weeks. In equation form, this says that

$$\text{desired production} = \frac{\text{Factory Order Backlog}}{\text{TARGET PRODUCTION DELAY}}$$

where the Factory Order Backlog is measured in units of the item being produced, and the TARGET PRODUCTION DELAY is measured in weeks.

In typical realistic factory settings, production cannot be instantaneously changed in response to variations in orders because it takes time to change production resources, such as personnel and equipment. A more complex model of production would include explicit consideration of each of these factors, but we will approximate them here by saying that there is an average delay of TIME TO ADJUST PRODUCTION before the actual factory production is brought into line with desired production.

In most realistic settings, the rate at which production can be adjusted varies depending on the immediate circumstances. Thus, the delay would not always be exactly equal to TIME TO ADJUST PRODUCTION. A simple model for this, but one which matches the data for many realistic settings, is that the time it takes to adjust production follows an exponential delay process. We will consider this particular approach in further detail below, but for now just consider the delay in bringing actual production into line with desired production to be variable with an average length of TIME TO ADJUST PRODUCTION.

The equations for the production process, as well as the rest of the first model for the production-distribution system, are shown in Figure 5.2b. Equation 12 of this figure shows that the TARGET PRODUCTION DELAY is 2 weeks, and equation 15 shows that the TIME TO ADJUST PRODUCTION is 4 weeks. (The values for these and other constants in the production-distribution model are illustrative and not intended to necessarily represent good management practice.)

Equation 2 shows that the desired production is given by the equation discussed earlier. Equation 4 shows that actual factory production is delayed from desired production by an average time of TIME TO ADJUST PRODUCTION. (In Vensim, the exponential delay function is called SMOOTH.)

When setting up a process model, it is important to keep the measurement units that you use consistent. If you use units of weeks of time in one place and months in another, then you will obviously get incorrect answers. (Note that you can set up a Vensim model to automatically check units. To simplify the presentation, we are not using this feature in our model.)

For the remainder of our discussion of the production-distribution system, we will focus on the retailer ordering process. We will assume that the production process has the characteristics just presented, and will not attempt to improve this. In reality, this could be a source of improvements. For example, just-in-time production processes are designed to improve the performance of a production process of this type.

```
(01) average retail sales
         = SMOOTH(retail sales, TIME TO AVERAGE SALES)
(02) desired production
         = Factory Order Backlog / TARGET PRODUCTION DELAY
(03) Factory Order Backlog
         = INTEG(retailer orders - factory production, 200)
(04) factory production
         = SMOOTH(desired production, TIME TO ADJUST PRODUCTION)
(05) FINAL TIME  = 50
(06) INITIAL TIME  = 0
(07) Retail Inventory = INTEG(factory production - retail sales, 400)
(08) retail sales = TESt input
(09) retailer orders = average retail sales
(10) SAVEPER  = TIME STEP
(11) TARGET PRODUCTION DELAY = 2
(12) TESt input = 100 + STEP(20, 10)
(13) TIME STEP  = 0.25
(14) TIME TO ADJUST PRODUCTION = 4
(15) TIME TO AVERAGE SALES = 1
```

Figure 5.2b *Vensim equations for first model*

Retailer Ordering

We now turn our attention to retail sales and orders to the factory by the retailer. We will assume that retail sales are predetermined. (That is, they are an *exogenous variable* to the portion of the process we are modeling.) We will soon consider what these orders are, but first we specify a procedure that the retailer uses to order from the factory.

The simplest ordering procedure is to order exactly what you sell. However, in practice, most retailers cannot instantly order each time they make a sale. Thus, ordering is based on an average over some time period. Furthermore, this average is likely to take into account recent trends. For example, if sales over the last few days have been up, then the retailer is likely to put more weight on that than on the lower sales of an earlier period.

A simple model of this type of averaging process is called *exponential smoothing*, and this will be studied in more detail later. For now, you can consider that this type of averaging is approximately taking an average over a specified period of time, but that more weight is given to recent sales than earlier sales. Thus, in the Figure 5.2a stock and flow diagram, the variable average retail sales is calculated by taking an exponential smooth of retail sales over the period TIME TO AVERAGE SALES. This is shown by equation 1 in the Figure 5.2b Vensim equations, and equation 15 shows that TIME TO AVERAGE SALES is equal to 1 week.

Note that the same function (called SMOOTH) is used for the exponential averaging process as was used for the exponential delay process shown for adjusting factory production in equation 4. It turns out that the equation for an exponential averaging process and an exponential delay process are identical, and thus the same function is used for these in Vensim. However, conceptually the two processes are somewhat different. For the delay, we are interested in how long it takes for something in the future to happen. For the average, we are interested in what the average was for some variable over a past period.

The retailer ordering equations are completed by equation 9 in Figure 5.2b, which says that retailer orders are equal to average retail sales.

Test Input

To complete this first model of the production-distribution system, we need to determine retail sales. Equation 8 of Figure 5.2b shows that these are equal to TESt input, and thus we need to specify this. Actual retail sales typically have some average value with random fluctuation around this average. There may also be seasonal variation and an overall trend, hopefully upward. Thus, your first thought is probably to use a complex test input which represents these features of the real world.

However, we will use a very simple test input, and it is important to understand why this particular input is used because it is often used as a test input for process simulation models. The input we use will be a simple step: The input will start at one level, remain constant at that level for a period, and then jump instantly to another level and remain constant at the new level for the remainder of the period studied. The implementation of this input is shown in equation 12 of Figure 5.2b. The function STEP is defined by the following equation:

$$\text{STEP(height step time)} = \begin{cases} 0 & \text{Time} < \text{step time} \\ \text{height} & \text{otherwise} \end{cases}$$

That is, the function is zero until the time is equal to step time, and then it is equal to height. Thus, equation 12 of Figure 5.2b says that TESt input is equal to 100 units per week until the time is 10 weeks, and then TESt input is equal to 120 units per week for the remainder of the time.

Why is this used as a test input? It seems quite unlikely that the actual sales would have this form! The reason for using this form of test input has to do with what we are trying to accomplish with our model, and thus we need to discuss the purpose of our modeling.

Our primary purpose in constructing this model is to determine ways to improve the performance of the production-distribution process. In particular, we are studying different possible retailer ordering policies and how these impact the performance of the entire production-distribution process. There are, of course, many different possible patterns of retail sales, and we want to make sure that the particular pattern that we use in our model allows us to study the characteristics of the process that are important to understand if we are to improve the performance of the process. Remarkably, a step pattern for the retail sales is a good pattern for this purpose.

Understanding in detail why this is true requires studying some theory that is beyond the scope of this text, but we will later look at a more realistic pattern of retail orders and show that the behavior of the production-distribution process in response to this more realistic pattern is remarkably similar to its behavior with a step input. If you have studied engineering systems, you have probably already learned that the response of a linear system to a step input completely characterizes the behavior of the system. While the processes that we are considering are generally nonlinear, their responses to a step input still gives important information about how the process responds to a variety of inputs.

To continue this theoretical discussion slightly longer, readers who have studied Fourier or Laplace analysis methods will remember that the frequency spectrum for a step function contains all frequencies. Therefore, using a step as input to a process excites all resonant frequencies of the process. These resonant frequencies are usually a critical determinant of the behavior of the process, and therefore the process response to a step input is often a good indicator of how the process will respond to a variety of inputs.

Other Model Equations

The remainder of the model equations in Figure 5.2b are mostly straightforward. Equations 3 and 7 for the stock variables Factory Order Backlog and Retail Inventory are known from the stock and flow diagram in Figure 5.2a, except for the initial values. We see from equation 3 that the initial value of Factory Order Backlog is 200 units, and equation 7 shows that the initial value of Retail Inventory is 400 units.

At an initial Factory Order Backlog of 200 units with a TARGET PRODUCTION DELAY of 2 weeks (as given by equation 11 in Figure 5.2b), the desired production is 200 2 = 100 units per week. As long as there is no variation in Factory Order Backlog, then factory production will be equal to desired production, and hence will also be equal to 100 units per week.

We see from equation 12 that the initial value of TESt input is 100 units per week, and hence from equation 8 this is also the initial value of retail sales. With no variation in retail sales, average retail sales will be equal to retail sales, and hence also equal to 100 units per week, and thus from equation 9 this will also be the retailer orders to the factory.

Since initial retail sales, and hence retailer order to the factory, are equal to factory production (100 units per week), then the system will initially be rather boring the factory will make 100 units per week, which will be sold by the retailer. The Factory Order Backlog will remain stable at 200 units, and the Retail Inventory will remain stable at 400 units.

When a process is in a situation like that described in the last few paragraphs where the variables remain constant over time, it is said to be in *equilibrium* or *steady state*. A steady state condition for a simulation model can be detected by examining the stocks in the model. In steady state, the sum of all inflows to each stock is equal to the sum of all outflows, and therefore the magnitudes of the stocks do not change over time.

If the production-distribution process we are studying had not started out in equilibrium, then even without any changes in TESt input some of the variables would have changed over time. For example, if the initial value for Factory Order

Backlog had been greater than 200 units, then this level would have declined over time *even if retail sales had remained steady at 100 units per week.* This is because at a Factory Order Backlog greater than 200 units factory production will exceed 100 units per week, which is the retailer order rate, and hence the flow out of Factory Order Backlog will exceed the flow in.

Since our purpose in this analysis is to study the impact of changes in retail sales on the production-distribution process, it is desirable to start the process model in steady state. Otherwise, it will be difficult to separate variations over time in the values of the various model variables which are due to changes in retail sales from those variations which are due to the lack of initial steady state. Similar arguments hold for many business process models, and it is usually good practice to initialize the variables in a model so that it starts in steady state.

The remaining equations in Figure 5.2b (equations 5, 6, 10, and 13) set characteristics of the simulation model. From equations 5 and 6 we see that the simulation will run for 50 weeks, or approximately one year. The rationale for setting the TIME STEP (equation 13) to 0.25 will discussed below.

5.2 Performance of the Process

Before reviewing Figure 5.2c which shows results from simulating the model in Figure 5.5a and Figure 5.2b, you may wish to consider how you expect the process to respond to the TESt input. This is a much simpler production-distribution system than many in the real world, and many of those real world systems are managed with relatively little analysis. Perhaps all this analysis is not necessary. What do you think will happen in the process? Is the retailer ordering policy used in this model a good one? What are its strengths and weaknesses?

Figure 5.2c shows plots of three key variables (retail sales, retailer orders, and Retail Inventory) when the simulation model in Figure 5.2b is run. For the first ten weeks everything remains constant. The graphs show that retail sales are 100 units per week, and retailer orders also remain at 100 units per week. Retail Inventory remains at 400 units. At week 10, retail sales jump to 120 units per week, and remain there for the remaining 40 weeks shown in the graph. Retailer orders do not immediately jump to 120 units per week because an average of past sales is used as a basis for ordering, and it takes a while for the average to climb to 120 units. However, since the averaging period (TIME TO AVERAGE SALES) is only 1 week, retailer orders quickly move upward, and by week 14 these are also at 120 units per week.

A careful reader may wonder why these orders do not reach 120 units per week by week 11 since the TIME TO AVERAGE SALES is only 1 week. An exponential averaging process actually considers a period longer than the averaging time, but it gives increasingly less weight to earlier values as time goes on. This will be discussed further below.

The behavior of retail sales and retailer orders is what we would probably have expected. What about Retail Inventory? Figure 5.2c shows that this remains level until week 10 and then starts to decline. It wiggles somewhat before leveling off at 340 units. It reaches a low of about 310 units at week 17, and then increases

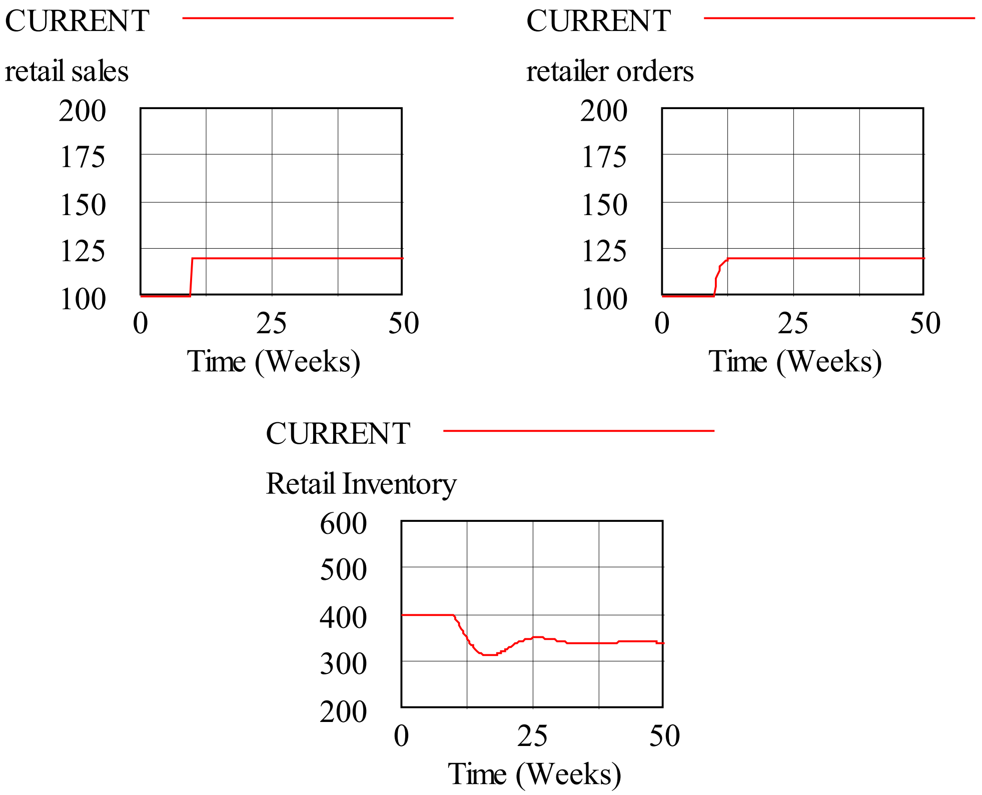


Figure 5.2c *Plots for first model*

to a peak of about 350 units at week 26 before dropping back to 340 units. Note that there is also a slight valley at around week 35.

This type of wiggling is called *oscillation*. When you did your intuitive prediction of how the process would perform, did you expect this oscillation? For that matter, did you predict that Retail Inventory would decline? While not shown in Figure 5.2c, after seeing this figure you will probably not be surprised to learn that Factory Order Backlog and factory production also both show similar oscillation to those shown by Retail Inventory.

While these oscillations are not large, they pose some challenges to a factory manager. Decisions have to be made about how to provide the necessary resources under oscillating production conditions. For example, do you lay off factory workers when production dips? Also, the revenue stream associated with oscillating conditions is likely to be uneven, which is generally not desirable.

We will shortly be paying considerably more attention to oscillations, but for now focus on the final level of Retail Inventory. Is this acceptable? For the particular retail sales stream analyzed here, it may be. In fact, the lower Retail Inventory resulting from the jump in sales probably means lower inventory carrying costs, and hence higher profit. However, a little additional thought shows that this could be a dangerous policy in some situations. In particular, suppose that retail sales continue to grow (as we would hope they do!). What happens then? In that case, because average retail sales will always be somewhat less than current retail sales, there will never be quite enough ordered to replace what is sold, and eventually Retail Inventory will be depleted.

This effect can be reduced by reducing the TIME TO AVERAGE SALES, which corresponds to more rapidly ordering, however, it cannot be entirely eliminated because even if you instantaneously order after each sale, you will still fall behind because of delays in production.

This effect is one reason that many production-distribution systems are moving to automated, speeded-up ordering systems. For example, Wal Mart has made extensive use of such systems in its rise to retailing dominance. However, there is a limit to what is possible along these lines, particularly in businesses where the supply chain is not yet highly integrated. Is there some approach that a retailer can use in ordering that will reduce the danger of running out of inventory when sales rise? (Incidentally, note that if sales steadily fall, then Retail Inventory will steadily rise.)

5.3 The Second Model

It seems that we need to directly consider the level of Retail Inventory when retailer orders are placed in order to make sure that this inventory does not reach undesirable levels. The stock and flow diagram in Figure 5.3a shows an approach to doing this. This is modified from the Figure 5.2a diagram as follows: There is an information arrow from Retail Inventory to retailer orders to show that these orders depend on Retail Inventory. There are two auxiliary constants DESIRED INVENTORY and TIME TO ADJUST INVENTORY which also influence retailer orders. The remainder of this diagram is the same as Figure 5.2a.

The approach to considering Retail Inventory in retailer orders which is shown in this diagram makes intuitive sense: There is a specified level of DESIRED INVENTORY and retailer orders are adjusted to attempt to maintain this level. Of course, we do not want to radically change our orders for every small change in Retail Inventory, and so we take some time to make the adjustment (TIME TO ADJUST INVENTORY). Turning this into a specific equation, there is now a component of retailer orders as follows:

$$\frac{\text{DESIRED INVENTORY} - \text{Retail Inventory}}{\text{TIME TO ADJUST INVENTORY}}$$

That is, if everything were to remain the same, the difference between DESIRED INVENTORY and the actual Retail Inventory would be eliminated in a time period equal to TIME TO ADJUST INVENTORY. Note that if DESIRED INVENTORY is below Retailer Inventory, then the ordering level will be reduced, while if DESIRED INVENTORY is above Retailer Inventory the ordering level will be increased.

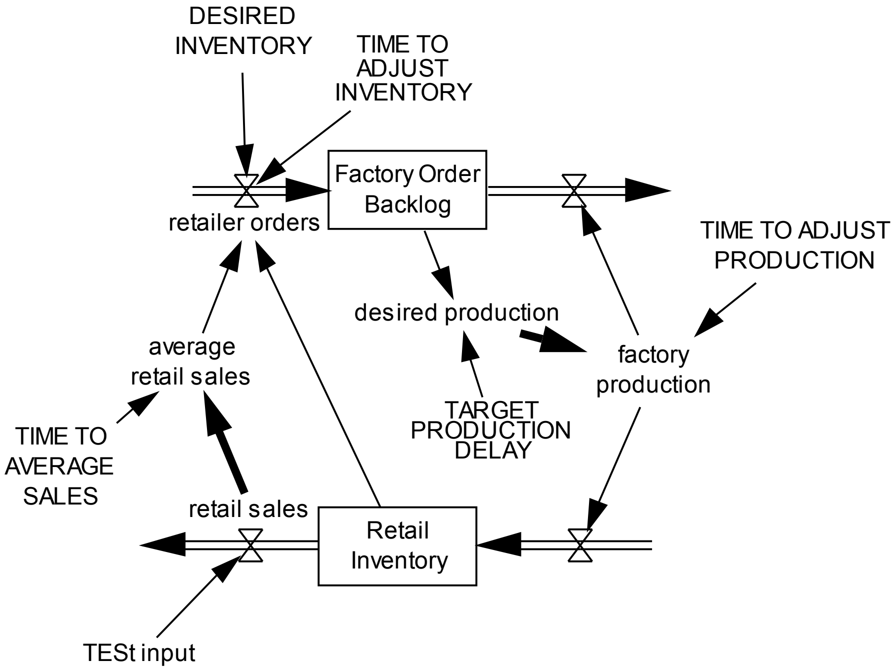


Figure 5.3a *Stock and flow diagram for second model*

The equations for the second model are shown in Figure 5.3b. These are identical to the equations in Figure 5.2b except that definitions have been added for the two constants DESIRED INVENTORY and TIME TO ADJUST INVENTORY, and the equation for retailer orders has been modified as discussed in the preceding paragraph. Specifically, in Figure 5.3b, equation 2 shows that the DESIRED INVENTORY is 400 units (which was the initial level of Retail Inventory), and equation 15 shows that the TIME TO ADJUST INVENTORY is 2 weeks. Equation 10 shows that retailer orders now include the component discussed above to adjust the level of Retailer Inventory, in addition to the component to replace retail sales.

A note is in order here on possible deficiencies in the formulation of this model. Because the component of the retailer orders equation (equation 10) to replenish inventory can have a negative sign, it is possible that the overall value of retailer orders could become negative. Exactly what would happen in that case depends on the ordering arrangements. It may be possible to withdraw previously placed orders, or it may not be possible to do this. In a more complete model, this would be taken into account. Our simple model assumes that it is possible to withdraw previously placed orders. In fact, the model even assumes that it is possible to withdraw more orders than you have actually placed, which is not likely to be true in a realistic setting.

```
(01) average retail sales = SMOOTH(retail sales, TIME TO AVERAGE SALES)
(02) DESIRED INVENTORY = 400
(03) desired production = Factory Order Backlog / TARGET PRODUCTION DELAY
(04) Factory Order Backlog
          = INTEG(retailer orders - factory production, 200)
(05) factory production
         = SMOOTH(desired production, TIME TO ADJUST PRODUCTION)
(06) FINAL TIME  = 50
(07) INITIAL TIME  = 0
(08) Retail Inventory = INTEG(factory production - retail sales, 400)
(09) retail sales = TESt input
(10) retailer orders = average retail sales
  + (DESIRED INVENTORY - Retail Inventory) / TIME TO ADJUST INVENTORY
(11) SAVEPER  = TIME STEP
(12) TARGET PRODUCTION DELAY = 2
(13) TESt input = 100 + STEP(20,10)
(14) TIME STEP  = 0.25
(15) TIME TO ADJUST INVENTORY = 2
(16) TIME TO ADJUST PRODUCTION = 4
(17) TIME TO AVERAGE SALES = 1
```

Figure 5.3b *Vensim equations for second model*

Thinking further along those lines, you will see that there is also no constraint in the model on the possible values for Factory Order Backlog and Retail Inventory. Thus, it is possible in this model for these to become negative. Again, a more complete model should take these issues into consideration. However, we are interested at the moment in the general characteristics of the performance of this process, rather than the details. The model we have developed will be sufficient for this purpose, as we will shortly see.

What do you think will be the performance of the modified ordering policy? Do you think it will cure the problem of under ordering when sales are growing and over ordering when sales are declining?

Figure 5.3c gives the answer, and it is not pleasant. The same three variables are plotted here as in Figure 5.2c which we considered earlier. We see that the system goes into large and growing oscillation. In fact, the situation is worse than it may first appear because the scales for some of the graphs is Figure 5.3c (as shown on the left side of the graphs) are greater than those for Figure 5.2c. Thus, the oscillations are greater than it might first appear by visually comparing Figure 5.2c and Figure 5.3c.

Retail sales still display the same step pattern as in Figure 5.2c. (They must do this since they are defined exogenously to the model.) However, both retailer orders and Retail Inventory wildly oscillate. Furthermore, both of these go substantially negative. As noted above, the model equations are probably not valid when this happens, so a real world system would not display exactly the

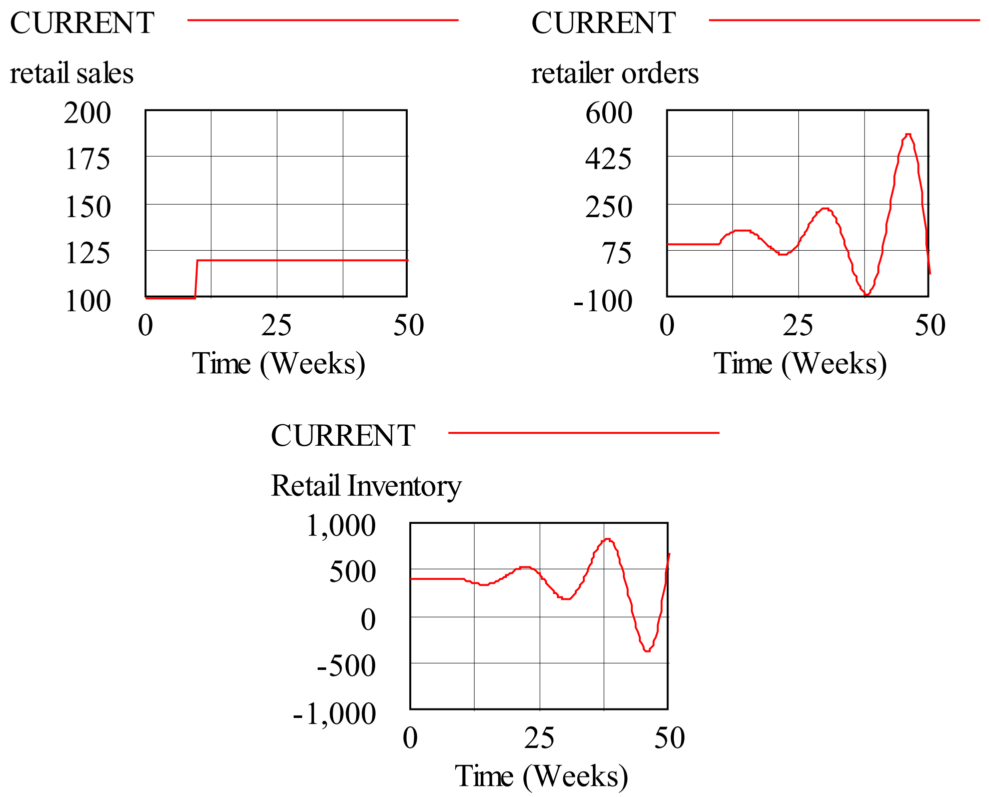


Figure 5.3c *Plots for second model*

behavior shown here. However, it is clear that the revised retailer ordering policy is highly unacceptable.

What is going wrong? Some thought shows the answer and provides insight into the performance of real world production-distribution systems. The difficulty with the new ordering policy results from the delays in producing the stock that has been ordered. It takes time for the orders to work their way through the Factory Order Backlog, be produced, and flow into Retail Inventory. While this is going on, retailer orders continue to be high in a continuing attempt to replace the declining Retail Inventory. Then, when the ordered units finally start to arrive, Retail Inventory grows. At this point, the inventory correction term in retailer orders turns negative in an attempt to reduce the level of Retail Inventory. Once again, it takes time for the impact of this to work its way through the process, and this eventually leads to an overcorrection in the opposite direction. Figure 5.3c shows this problem getting worse as time goes on.

A Technical Note

Our main emphasis is on model formulation, but a brief note is in order on how the model equations are solved to develop a graph like the one shown in Figure 5.3c. As our earlier development showed, the solution of the model equations requires that some integrals be calculated. There are a variety of different methods to do this, and most simulation packages provide options for how this is done. The simplest of these methods, called Euler integration, was used to determine Figure 5.3c (as well as all the other output presented in this text).

The Euler integration method implemented in Vensim consists of the following steps:

1. Set Time to its initial value.
2. Set all of the stocks in the model to their initial values as specified by the initial value argument of the INTEG function for each stock.
3. Compute the rate of change at the current value of Time for each stock by computing the net values of all the flows flow into and out of each stock. (That is, flows into a stock *increase* the value of the stock, while flows out of a stock *decrease* the value of the stock.)
4. Assume that the rate of change for each stock will be constant for the time interval from Time to $\text{Time} + \text{TIME STEP}$, and compute how much the stock will change over that interval. This can be expressed in equation form as follows: If the rate of change for a particular stock calculated in Step 3 is rate(Time), then the value of that stock at time $\text{Time} + \text{TIME STEP}$ is given by

$$\text{Stock}(\text{Time} + \text{TIME STEP}) = \text{Stock}(\text{Time}) + \text{TIME STEP} \times \text{rate}(\text{Time})$$

5. Add TIME STEP to Time.
6. Repeat Steps 3 through 5 until Time reaches FINAL TIME.

From this procedure, it is apparent that the accuracy of the Euler method is influenced by the value chosen for the model constant TIME STEP. It is generally recommended that a value of TIME STEP be selected that is less than one-third of the smallest time-related constant in the model. In the Figure 5.3b model, the smallest such constant is TIME TO AVERAGE SALES which is equal to 1 week. Therefore, TIME STEP was set equal to 0.25, which is one-quarter of TIME TO AVERAGE SALES. A quick test for whether TIME STEP is small enough is to reduce it by a factor of two and rerun the simulation. If there is no significant change in the output, then this indicates that TIME STEP is small enough.

However, even with a small value of TIME STEP, the Euler integration method can yield inaccurate results when there are significant oscillations in a process. As Figure 5.3c shows, there are significant oscillations in the process we are studying. Therefore, other, more sophisticated integration procedures should be used if high accuracy is needed. However, even without using these more sophisticated integration method, it is clear that the ordering policy in our second model is not a good one.

More sophisticated integrated methods available in many simulation packages often include one or more of the Runge-Kutta methods. The underlying idea of these methods is to estimate the rate at which the flows into stocks are varying over time, and then use this information to improve on the approximation in Step 4 of the Euler procedure presented above. When this is done, it is often possible to achieve improved accuracy without as much increase in computation as would be necessary if the value of TIME STEP were decreased in the Euler procedure.

5.4 The Third Model

You might suspect that the problem displayed in Figure 5.3c is due to including a component in the retailer orders to take account of retail sales. Perhaps if we focus completely on Retail Inventory when making retailer orders, this will fix the problem. The stock and flow diagram in Figure 5.4a shows this approach. This differs from the diagram in Figure 5.3a in that the variables related to ordering to replace retail sales are removed. Specifically, at the left center of the diagram, the auxiliary variable average retail sales is removed along with the constant TIME TO AVERAGE SALES.

The corresponding equations are shown in Figure 5.4b. These differ from the equations in Figure 5.3b in that the equations for average retail sales and TIME TO AVERAGE SALES are removed, and the term for average retail sales is removed from equation 9 for retailer orders.

The resulting performance is shown in Figure 5.4c, and we see that this performance is even worse than what was shown in Figure 5.3c. (Note that the scales for some of the graphs in Figure 5.4c are considerably increased relative to Figure 5.3c, and thus the amplitude of the oscillations are much worse.)

Clearly this is not the answer.

5.5 The Fourth Model

Return now to the second model, whose performance is shown in Figure 5.3c. While the oscillations are clearly unacceptable, there is one way in which the performance of this process is better than the performance for the first model which was shown in Figure 5.2c. While there are wild oscillations in the Retail Inventory in Figure 5.3c, these oscillations are around an average level of 400 units, which is the Retail Inventory that we are trying to maintain. Thus, this process does not display the declining Retail Inventory level that is shown in Figure 5.2c. Unfortunately, the oscillations shown in Figure 5.3c are much too great to be acceptable in most real world production-distribution systems.

Our discussion above of the second model showed that the oscillations are due to the delays in obtaining the product that the retailer has ordered. This is sometimes referred to as a pipeline effect. We place orders into the supply

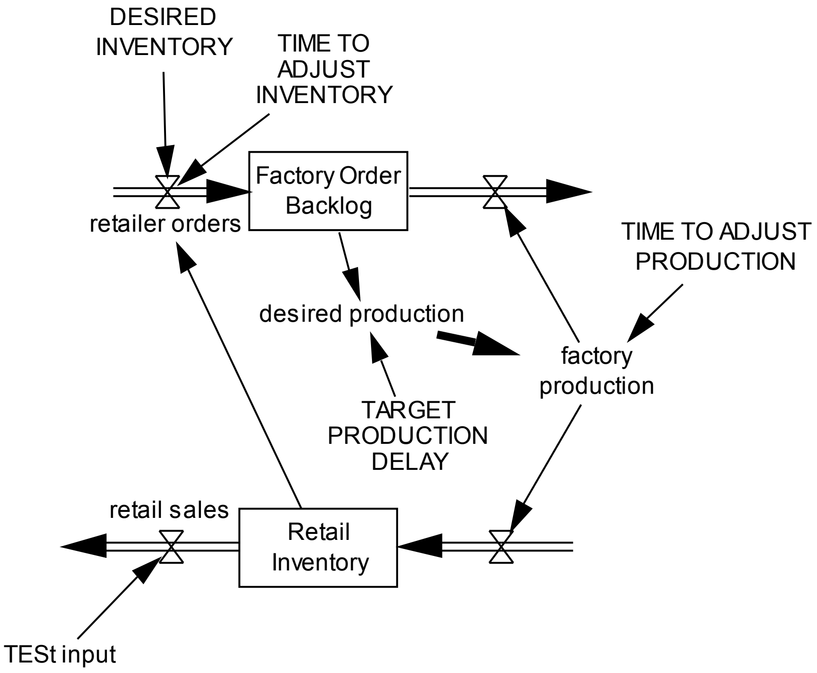


Figure 5.4a *Stock and flow diagram for third model*

```
(01) DESIRED INVENTORY = 400
(02) desired production
          = Factory Order Backlog / TARGET PRODUCTION DELAY
(03) Factory Order Backlog
          = INTEG(retailer orders - factory production, 200)
(04) factory production
          = SMOOTH(desired production, TIME TO ADJUST PRODUCTION)
(05) FINAL TIME  = 50
(06) INITIAL TIME  = 0
(07) Retail Inventory = INTEG(factory production - retail sales, 400)
(08) retail sales = TESt input
(09) retailer orders
          = (DESIRED INVENTORY - Retail Inventory)
             / TIME TO ADJUST INVENTORY
(10) SAVEPER  = TIME STEP
(11) TARGET PRODUCTION DELAY = 2
(12) TESt input = 100 + STEP(20, 10)
(13) TIME STEP  = 0.25
(14) TIME TO ADJUST INVENTORY = 2
(15) TIME TO ADJUST PRODUCTION = 4
```

Figure 5.4b *Vensim equations for third model*

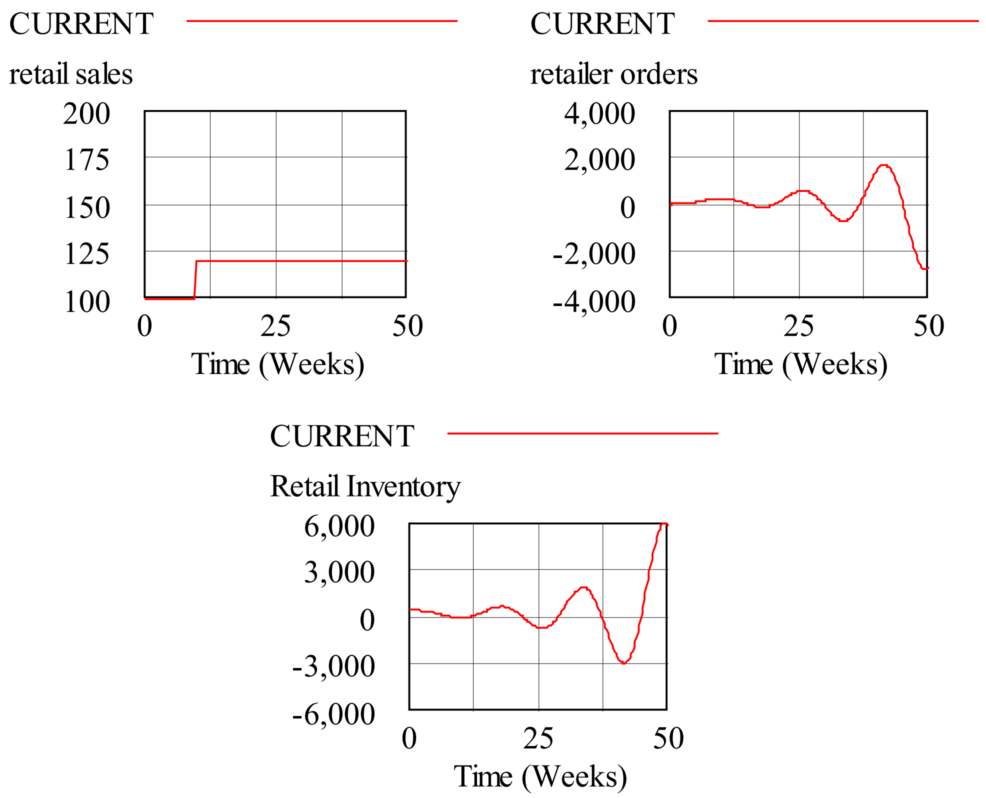


Figure 5.4c *Plots for third model*

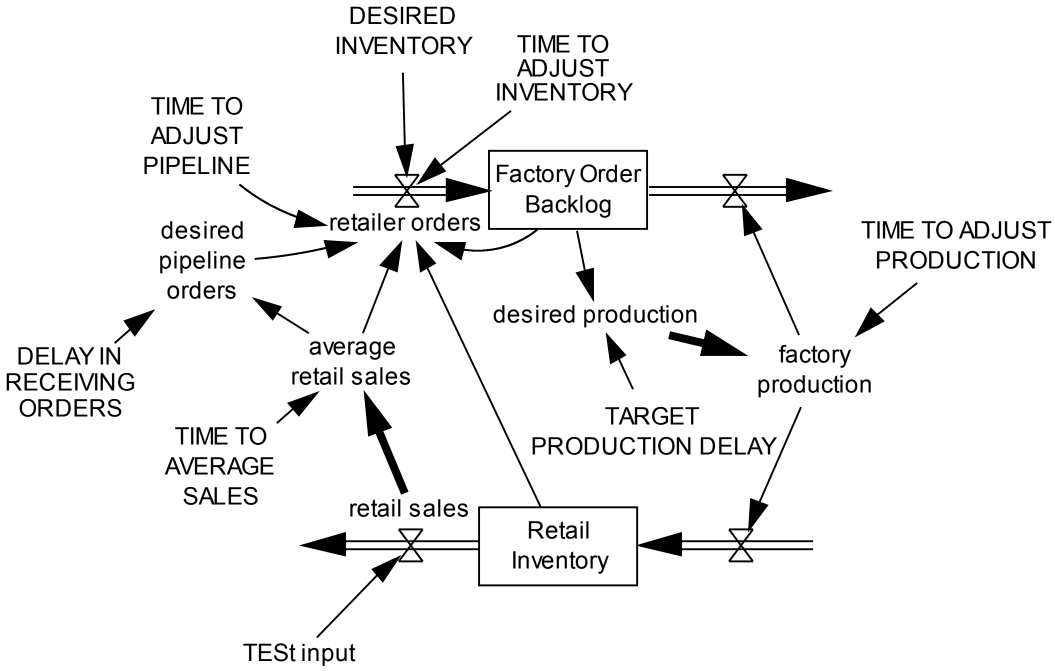


Figure 5.5a *Stock and flow diagram for fourth model*

pipeline, and then we basically forget about these orders and keep ordering. As discussed above, this leads to the oscillatory performance of the process.

The stock and flow diagram in Figure 5.5a shows one way to account for the orders that are in the pipeline. This diagram is developed from the diagram in Figure 5.3a (not Figure 5.4a!) as follows: An information arrow is added from Factory Order Backlog to retailer orders. Thus, the retailer ordering policy will now explicitly take into account the backlog of orders. How this is done is indicated by the new auxiliary variable desired pipeline orders in the center of the left side of the diagram, and the two new constants DELAY IN RECEIVING ORDERS and TIME TO ADJUST PIPELINE.

The desired pipeline orders are the amount we want to have on order at any time, and this depends on average retail orders and the DELAY IN RECEIVING ORDERS. In particular, we need to have an amount in the supply pipeline equal to the product of average retail orders and DELAY IN RECEIVING ORDERS if we are to continue to receive enough to replenish our sales on average. That is,

$$\begin{aligned}\text{desired pipeline orders} &= \text{average retail sales}\\ &\quad \times \text{DELAY IN RECEIVING ORDERS}\end{aligned}$$

However, following the same logic presented above regarding adjusting retailer orders to account for changes in Retail Inventory, we do not want to instantly change our orders in response to changes in desired pipeline orders. Hence there is a constant TIME TO ADJUST PIPELINE which plays a similar role to TIME TO ADJUST INVENTORY. Therefore, there should be a component in the retailer order equation to account for orders in the supply pipeline as follows:

$$\frac{\text{desired pipeline orders} - \text{Factory Order Backlog}}{\text{TIME TO ADJUST PIPELINE}}$$

The required model equations are shown in Figure 5.5b. The value of DELAY IN RECEIVING ORDERS is set to agree with TARGET PRODUCTION DELAY. Since the factory has set up a TARGET PRODUCTION DELAY of 2 weeks (equation 14), DELAY IN RECEIVING ORDERS is also set to this value in equation 2. The TIME TO ADJUST PIPELINE is also set to 2 weeks in equation 18. Finally, the additional retailer ordering term discussed above is added to retailer orders in equation 12.

The resulting process performance is shown in Figure 5.5c. This is substantially improved relative to what is shown in any of the earlier figures. The magnitude of oscillation for Retail Inventory is not much greater than in the original process in Figure 5.2c, but now the Retail Inventory fairly quickly returns to the desired level of 400 units. (Note that the scales for Figure 5.2c and Figure 5.5c are the same.) Retailer orders now rise above retail sales, but they then quickly drop back to the level of retail sales without the wild oscillations that were displayed in the second and third models. (This type of behavior is called an *overshoot*.) This performance is pretty good, although there is still some oscillation in Retail Inventory.

5.6 The Fifth Model

After some study of the fourth model, you might consider a possible enhancement to reduce the amount of oscillation. In the fourth model, a constant value is used for the DELAY IN RECEIVING ORDERS. Perhaps adding a forecast for this delay would improve the performance of the process. Figure 5.6a shows a stock and flow diagram for a process which includes such a forecast. The constant DELAY IN RECEIVING ORDERS shown in Figure 5.5a has been replace by an auxiliary variable delivery delay forecast by retailer, which is shown in the upper left corner of the diagram. This forecast depends on a constant TIME TO DETECT DELIVERY DELAY and another auxiliary variable delivery delay estimate. The delivery delay estimate depends on Factory Order Backlog and factory production.

Note that a new type of diagram element has been introduced. This is the symbol for factory production in the upper right portion of the diagram. This is a second copy of factory production. (The first copy of this variable is shown in the center right of the diagram.) The second copy is shown in angle brackets (<, >), and such a second copy of a variable is called a *shadow variable* or a *ghost variable*. Shadow variables are used to avoid the need to run additional arrows in a stock and flow diagram which would make it more complex and confusing

```
(01) average retail sales = SMOOTH(retail sales,TIME TO AVERAGE SALES)
(02) DELAY IN RECEIVING ORDERS = 2
(03) DESIRED INVENTORY = 400
(04) desired pipeline orders
          = DELAY IN RECEIVING ORDERS * average retail sales
(05) desired production = Factory Order Backlog / TARGET PRODUCTION DELAY
(06) Factory Order Backlog
          = INTEG(retailer orders - factory production, 200)
(07) factory production
          = SMOOTH(desired production, TIME TO ADJUST PRODUCTION)
(08) FINAL TIME  = 50
(09) INITIAL TIME  = 0
(10) Retail Inventory = INTEG(factory production - retail sales, 400)
(11) retail sales = TESt input
(12) retailer orders = average retail sales
  + (DESIRED INVENTORY - Retail Inventory) / TIME TO ADJUST INVENTORY
  + (desired pipeline orders - Factory Order Backlog)
             / TIME TO ADJUST PIPELINE
(13) SAVEPER  = TIME STEP
(14) TARGET PRODUCTION DELAY = 2
(15) TESt input = 100 + STEP(20,10)
(16) TIME STEP  = 0.25
(17) TIME TO ADJUST INVENTORY = 2
(18) TIME TO ADJUST PIPELINE = 2
(19) TIME TO ADJUST PRODUCTION = 4
(20) TIME TO AVERAGE SALES = 1
```

Figure 5.5b *Vensim equations for fourth model*

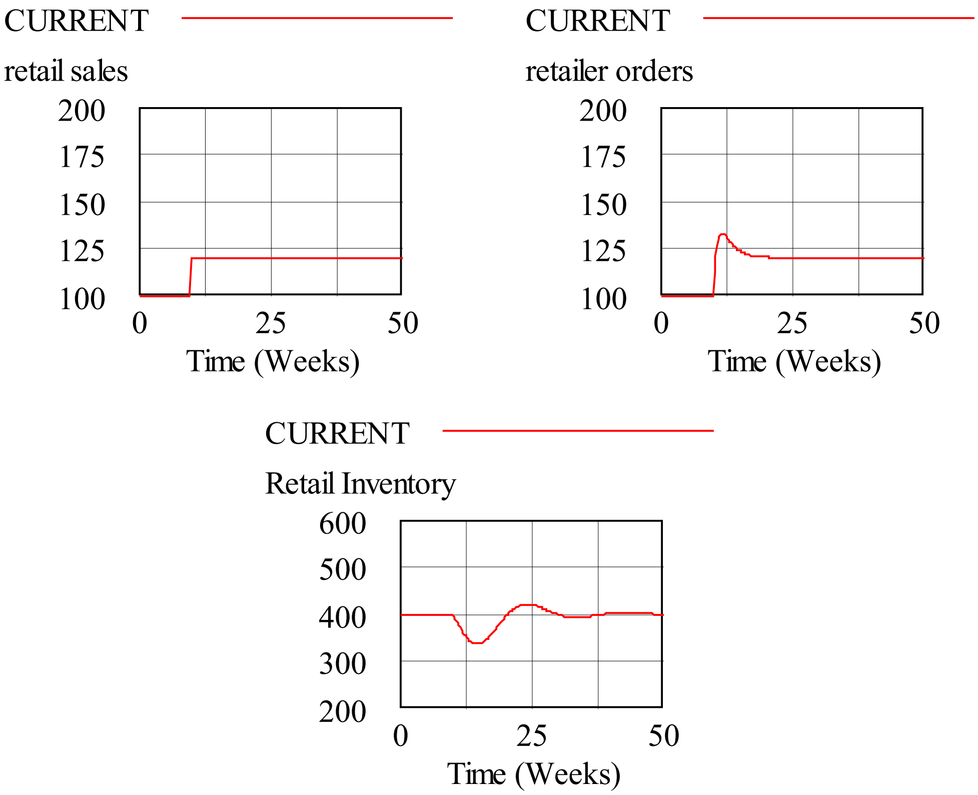


Figure 5.5c *Plots for fourth model*

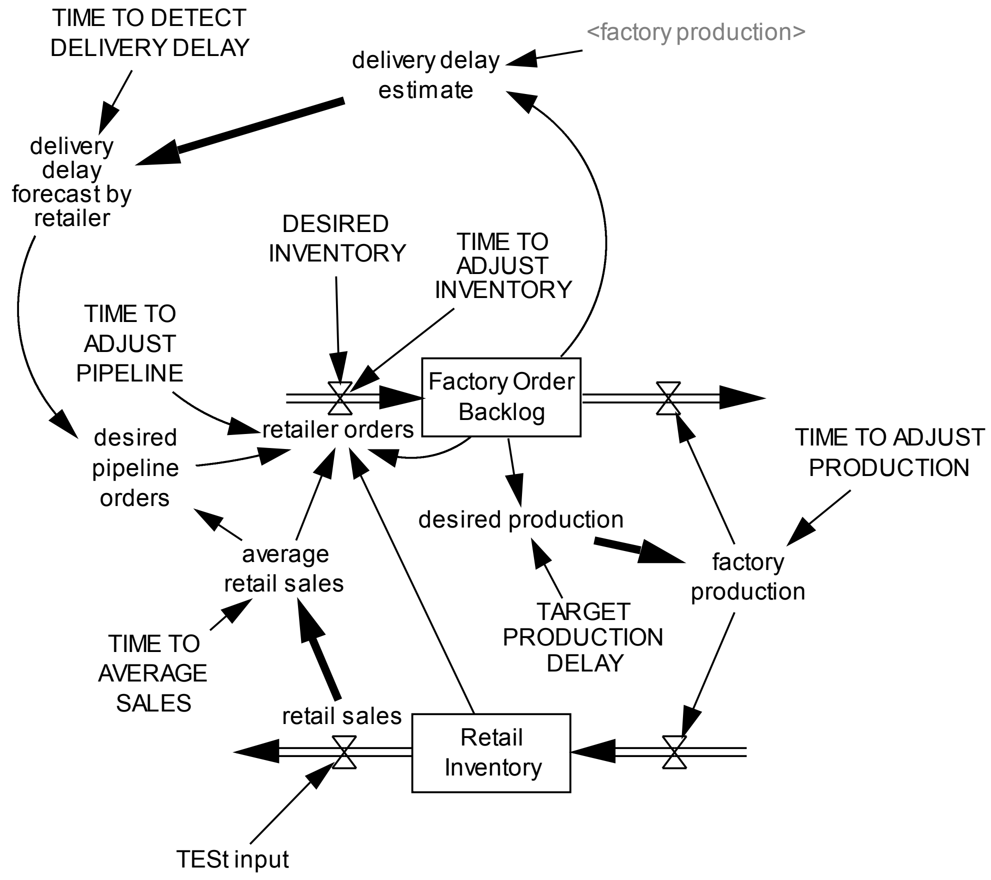


Figure 5.6a *Stock and flow diagram for fifth model*

to read. In this case, it avoids the need to run an information arrow from the original version of factory production in the right center of the diagram all the way to the top center of the diagram where the delivery delay estimate variable is located.

The forecasting submodel is an attempt to model what an actual retailer might be able to do to forecast what is happening at its supplier. Such a retailer is likely to have some idea of what Factory Order Backlog and factory production are at any time. The product of these yields an estimate of delivery delay:

$$\text{delivery delay estimate} = \frac{\text{Factory Order Backlog}}{\text{factory production}}$$

However, the retailer s estimates of Factory Order Backlog and factory production are likely to be somewhat out of date at any given time, and also influenced by what has been happening at the factory over a period of time. A simple model of this is the exponential averaging process that we have briefly discussed earlier. Thus, the delivery delay forecast by retailer is modeled as an exponential average of the delivery delay estimate over an averaging period of TIME TO DETECT DELIVERY DELAY.

The equations for the fifth model are shown in Figure 5.6b. Equation 2 gives delivery delay estimate, and equation 3 gives delivery delay forecast by retailer. The constant TIME TO DETECT DELIVERY DELAY is given in equation 22 as 2 weeks.

Alas, as Figure 5.6c shows, adding this forecast makes things somewhat worse than in the fourth model. The process now oscillates. The basic problem is that forecasts tend to predict that current trends will continue into the future. Thus, when retail sales jump at 10 weeks, the forecast leads to more extreme over ordering than in the fourth model. This overcorrection problem also occurs during the later attempt to reduce ordering, and the oscillating process continues. Forecasts sometimes do not help system performance, as this example shows.

5.7 Random Order Patterns

At this point, some readers may say, Yah, but this model is too simple. The real world is more complex than this, and things average out. You don t really have to worry about all this stuff in the real world. While this is a natural reaction, it is a little strange when you think about it: A more complicated process will perform better and be easier to manage? This doesn t seem very likely. And the data doesn t support that view. The oscillatory behavior of production-distribution systems, as well as many other social-technical systems (including the national and world economies) is well documented.

As a small confirmation of the more general applicability of what we have seen in this chapter, Figure 5.7 shows the performance of the second model and the fourth model that we studied above in the presence of random retail orders. To produce these diagrams, equation 12 of the second model (shown in Figure 5.3b) and the equivalent equation 16 of the fourth model (shown in Figure 5.5b) were replaced by

$$\text{TESt input} = 100 + \text{STEP}(20\ \ 10) * \text{RANDOM UNIFORM}(0\ \ 1\ \ 0)$$

The Vensim function RANDOM UNIFORM(m, x, s) produces random numbers that are uniformly distributed between m and x, with the argument s (called the *seed*) setting the specific stream of random numbers. Therefore, this modified equation will produce a TESt input of 100 until week 10, and then it will produce a random TESt input that is distributed uniformly between 100 and 120.

Note that some of the scales are different for Figure 5.7a and Figure 5.7b. When we compare these graphs to the corresponding Figure 5.3c and Figure 5.5c, we see that the performance is very similar with a random retail order stream to what was seen with a step order stream. This supports the statement made earlier that the step input often serves as a good test input for a process model. Furthermore, these results support a conclusion that the oscillations in the process are due to innate characteristics of the process and not to external characteristics of, for example, the retail order stream.

```
(01) average retail sales = SMOOTH(retail sales, TIME TO AVERAGE SALES)
(02) delivery delay estimate = Factory Order Backlog / factory production
(03) delivery delay forecast by retailer
     = SMOOTH(delivery delay estimate, TIME TO DETECT DELIVERY DELAY)
(04) DESIRED INVENTORY = 400
(05) desired pipeline orders
     = delivery delay forecast by retailer * average retail sales
(06) desired production = Factory Order Backlog / TARGET PRODUCTION DELAY
(07) Factory Order Backlog
     = INTEG(retailer orders - factory production, 200)
(08) factory production
     = SMOOTH(desired production,TIME TO ADJUST PRODUCTION)
(09) FINAL TIME  = 50
(10) INITIAL TIME  = 0
(11) Retail Inventory = INTEG(factory production - retail sales,400)
(12) retail sales = TESt input
(13) retailer orders = average retail sales
  + (DESIRED INVENTORY - Retail Inventory) / TIME TO ADJUST INVENTORY
  + (desired pipeline orders - Factory Order Backlog)
      / TIME TO ADJUST PIPELINE
(14) SAVEPER  = TIME STEP
(15) TARGET PRODUCTION DELAY = 2
(16) TESt input = 100 + STEP(20,10)
(17) TIME STEP  = 0.25
(18) TIME TO ADJUST INVENTORY = 2
(19) TIME TO ADJUST PIPELINE = 2
(20) TIME TO ADJUST PRODUCTION = 4
(21) TIME TO AVERAGE SALES = 1
(22) TIME TO DETECT DELIVERY DELAY = 2
```

Figure 5.6b *Vensim equations for fifth model*

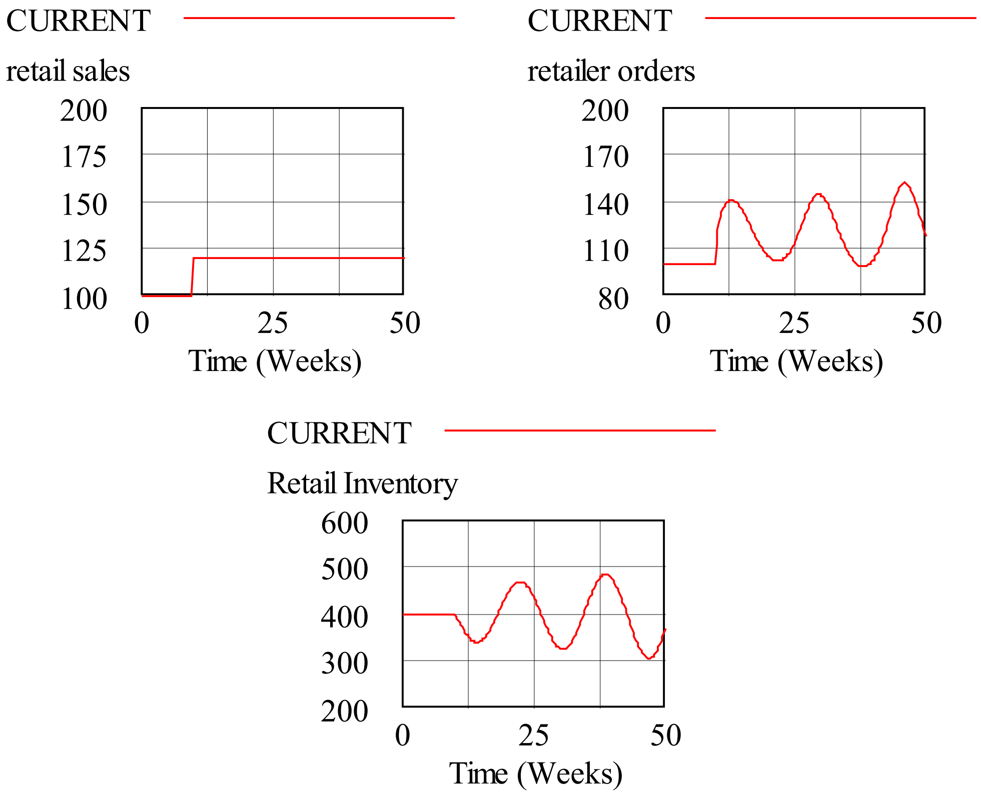


Figure 5.6c *Plots for fifth model*

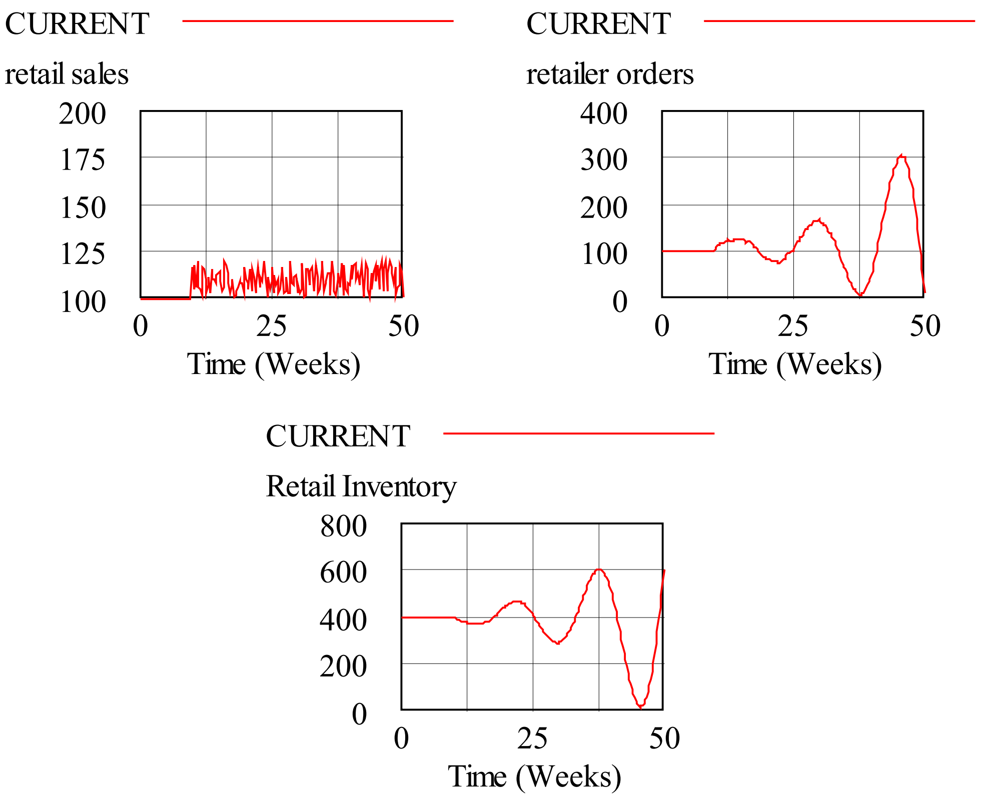


Figure 5.7a *Random retail sales for second model*

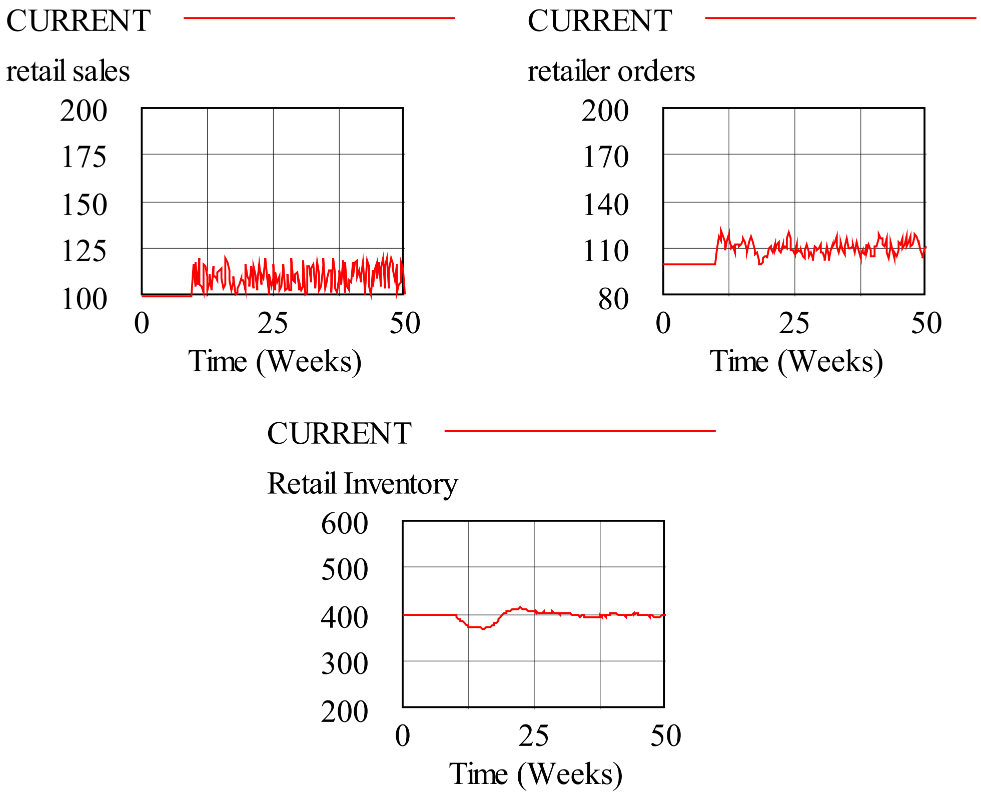


Figure 5.7b *Random retail sales for fourth model*

5.8 Concluding Comments

Production-distribution processes, and similar structures in service businesses, are widespread throughout industry. Understanding these processes is useful for any business person. The difficulty of controlling these processes which was displayed in this example is shared by many real world processes. The result in many such processes is a massive control structure to ensure stability. Unfortunately, such structures often make the processes strongly resistant to change when the external environment changes. In the remainder of this text, we will investigate ways of looking at processes that can help you in the search for better performance.

5.9 Reference

W. E. Jarmain (ed.), *Problems in Industrial Dynamics*, The MIT Press, Cambridge, Massachusetts, 1963.

CHAPTER 6

Delays, Smoothing, and Averaging

Delays are inherent in many management processes. It takes time to make a product or deliver a service. It takes time to hire or lay off workers. It takes time to build a new facility. In this chapter, we will investigate how such delays can be modeled. We will also investigate the related topic of *smoothing* (averaging) information flows.

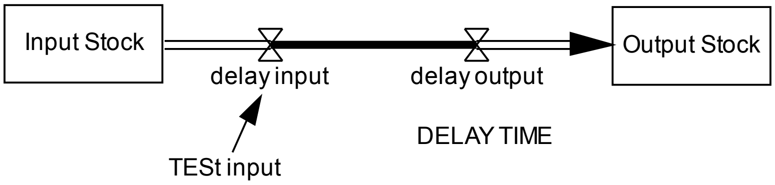
6.1 Pipeline Material Flow Delays

The simplest type of delay to visualize is the pipeline delay where material flows into one end of the delay and flows out the other end unchanged some specified period of time later, just like water flowing through a pipeline. Most simulation systems include one or more functions to implement a pipeline delay. For example, Vensim includes the DELAY FIXED and DELAY MATERIAL functions.

The use of the DELAY FIXED function is illustrated in Figure 6.1. The stock and flow diagram in Figure 6.1a demonstrates a suggested notation for indicating delays in a diagram. In this diagram, there is a delay in the flow from Input Stock to Output Stock. The flow out of Input Stock is controlled by the flow variable delay input, and this flow enters the delay leading from delay input to delay output, which controls the flow into Output Stock. The delay between these two flow controls is indicated by a thick solid pipe.

There are three arguments for the DELAY FIXED function, as shown in equation 2 of Figure 6.1b. The first is the input to the delay, the second is the delay time through the pipeline, and the third specifies what the output should be from the pipeline from the beginning of the simulation until there has been enough time for input to reach the end of the pipeline.

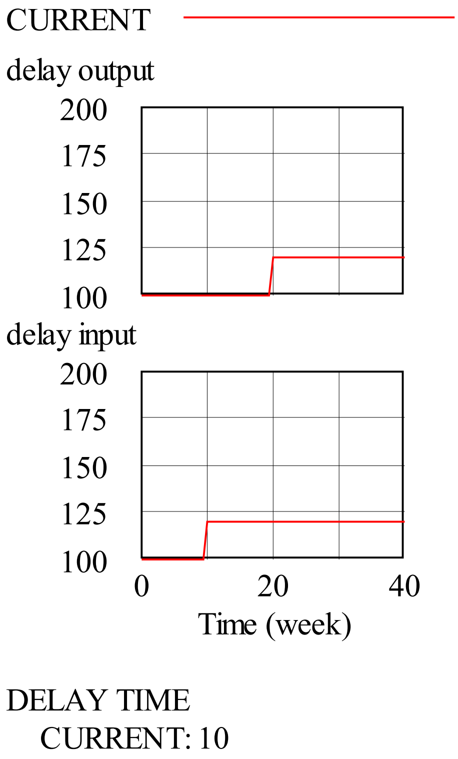
The results from a simulation run of the Figure 6.1b model are shown in Figure 6.1c. A step TESt input has been used, and we see from this figure that the delay output is the same step, but delayed by the length DELAY TIME of the pipeline, which is 10. Regardless of the form of the input, the output



a. Stock and flow diagram

```
(01) delay input = TESt input
(02) delay output
     = DELAY FIXED(delay input,DELAY TIME, delay input)
(03) DELAY TIME = 10
(04) FINAL TIME  = 40
(05) INITIAL TIME  = 0
(06) Input Stock = INTEG(delay input, 10000)
(07) Output Stock = INTEG(delay output, 0)
(08) SAVEPER  =  TIME STEP
(09) TESt input = 100 + STEP(20, 10)
(10) TIME STEP  = 0.5
```

b. Vensim equations



c. Delay

Figure 6.1 *Pipeline delay in material flow*

from the pipeline will be identical to the input, but delayed by the length of the pipeline.

In some business processes, the length of the delay in a pipeline delay can change depending on conditions elsewhere in the process. For example, the delay in a production process may change depending on the available personnel. If you need to model a situation of this type, then carefully consult the reference manual for your simulation package to be sure you understand what function to use to correctly model such a situation. In particular, you should make sure that the function you use conserves the material in the delay when the delay time changes. In Vensim, the DELAY MATERIAL function should be used for such a situation. This function requires you to specify some information about what happens to material flow when the delay changes.

6.2 Third Order Exponential Delays

The pipeline delay is an intuitively appealing model for material flow delays, but in some situations the flow process is not quite as clearcut. When orders are placed, sometimes all of the ordered goods do not arrive at exactly the same time. That is, there may be some variation in the delay time for the goods to arrive. The third order exponential delay provides a simple model for this situation. The use of this type of delay is illustrated in Figure 6.2. The same stock and flow diagram as in Figure 6.1a applies here, but the function for the delay between delay input and delay output is different than in Figure 6.1. The only change in the information you need to enter is in equation 2, and the modified equation is shown in Figure 6.2a.

The third order exponential delay equation in Vensim is called DELAY3, and it has two arguments. The first is the input variable to the delay, and the second is the delay time through the delay. The initial output from the delay is automatically set equal to the initial input to the delay. (Vensim provides another function DELAY3I which is identical to DELAY3, except that the initial output from the delay can be specified as different from the initial input.)

A step input to the third order exponential delay and the resulting output are shown in Figure 6.2b. In the left graph, a step input is shown, and the right graph shows that the output from this changes more gradually than with the pipeline delay. There is very little output for two or three time units after the step occurs, and then the output gradually comes through.

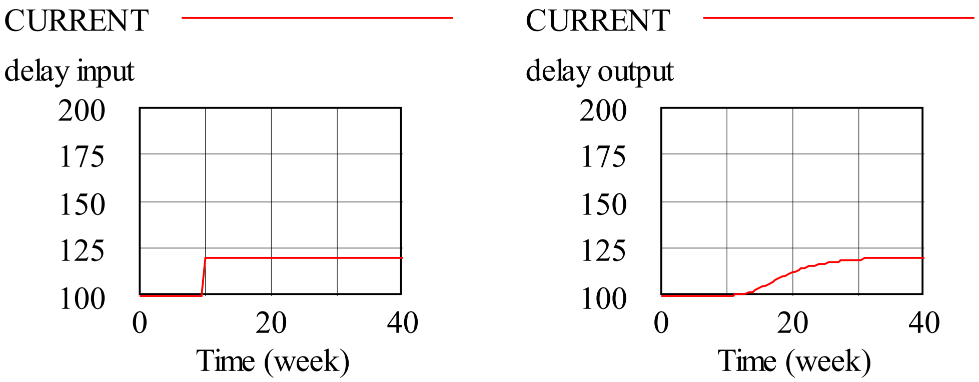
Material flow through a third order exponential delay is conserved even if the length of the delay changes while flow is occurring.

While the third order exponential delay is a satisfactory way to model many situations where there is some variability in the flow rate through the delay, you should be aware that it impacts time-varying inputs in a way that may not be appropriate as a model for some material flow processes. This is illustrated in Figure 6.3.

Figure 6.3a shows the changes from the Figure 6.2a equations which were made to generate the results shown in Figure 6.3b. Specifically, the step input of height 20 starting at time 10 was replaced with a sinusoidal input of amplitude

```
(01) delay input = TESt input
(02) delay output = DELAY3(delay input, DELAY TIME)
(03) DELAY TIME = 10
(04) FINAL TIME  = 40
(05) INITIAL TIME  = 0
(06) Input Stock = INTEG(delay input, 10000)
(07) Output Stock = INTEG(delay output, 0)
(08) SAVEPER  = TIME STEP
(09) TESt input = 100 + STEP(20, 10)
(10) TIME STEP  = 0.5
```

a. Vensim equations



b. Delay

Figure 6.2 *Third order delay in material flow*

20 starting at time 10. The results for three different cycle lengths (periods) for this sinusoidal are shown in Figure 6.3b. The left hand graphs in Figure 6.3b show the delay input for three different values of CYCLE LENGTH, and the right hand graphs show the resulting delay output for each of these three inputs.

Each of the three delay inputs has an amplitude of 20, but the cycle lengths varies. The top curve marked RUN4 has a cycle length of 4 time units, the middle curve marked RUN13 has a cycle length of 13, and the bottom curve marked RUN52 has a cycle length of 52. Thus, if the time units are weeks, these correspond to roughly monthly, quarterly, and annual cycles. Note that the three left hand graphs have the same amplitudes.

However, the amplitudes shown in the output curves in the right hand graphs are substantially different for the three inputs. The amplitude of the graph with a 4 week cycle is so small that it is hard to see much wiggle at all, the graph for the 13 week cycle has a somewhat larger amplitude, and the graph for the 52 week cycle has the largest amplitude. Note that none of these curves has as large an amplitude as the input. Furthermore, each of the output graphs is shifted somewhat to the right from the corresponding input graph. That is, the input is delayed as we would expect. (This delay is sometimes referred to as a *phase shift* when the input is a sinusoidal curve.)

These graphs illustrate that a third order exponential delay reduces (attenuates) inputs which vary over time, and furthermore, it attenuates inputs which vary faster more than inputs which vary slower. Using engineering jargon, a third order exponential delay differentially*filters* out higher frequency variations, as well as *phase shifting* the input.

6.3 Information Averaging

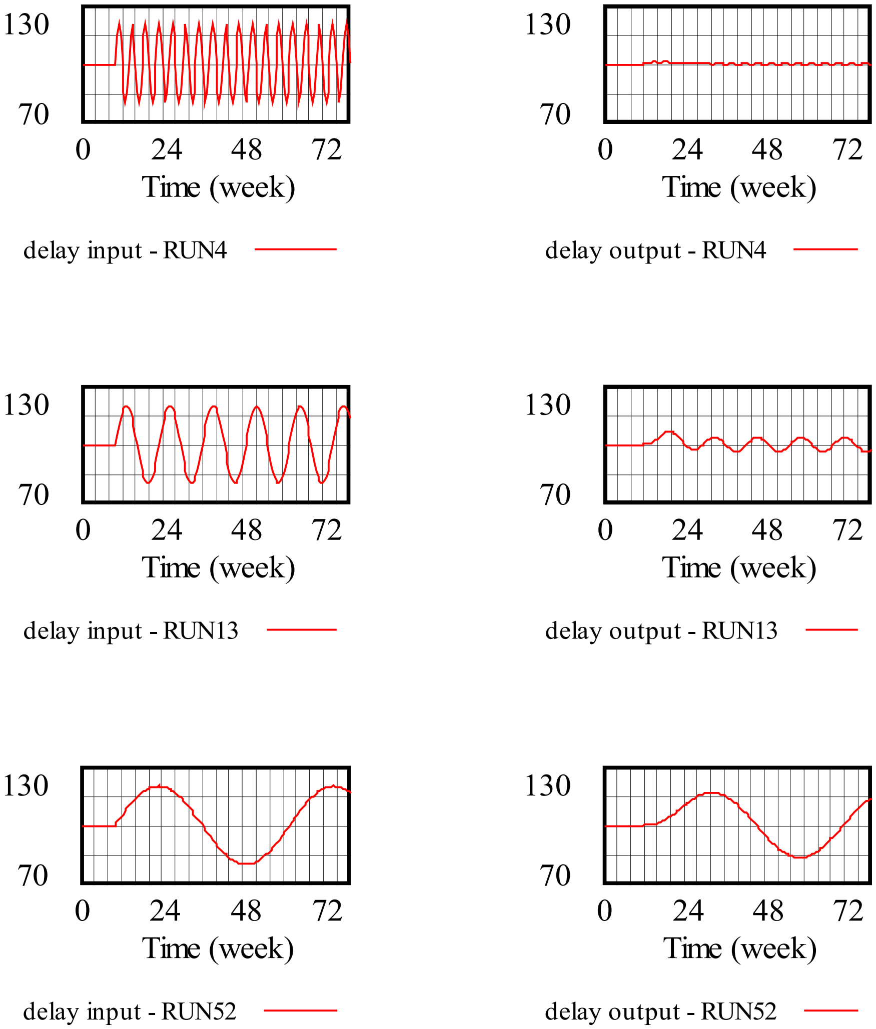
Many management actions should not be made in reaction to every random variation in the environment. For example, it takes time and resources to train workers or acquire large capital equipment. You do not want to undertake major changes in workforce or capital without some assurance that the long term conditions warrant this. In such situations, averaging is used to deal with the inherent irregularity of the processes.

The process of attempting to detect underlying, significant changes in data, while ignoring random, transitory fluctuations is called *averaging* or *smoothing.* This process can either be formal using statistical methods, or it can be informal and done on the fly while going about the daily activities of business management. This psychological smoothing is widespread, and experience with process models shows that it can have a significant impact on the dynamics of a business process. Therefore, it is important to model averaging/smoothing in a process.

In considering averaging, it is important to understand that the use of an average in decision making implicitly introduces a delay into the decision making. If, for example, you act based on the average of sales for the last month, rather than current sales, then you are inherently acting on delayed information since all of the numbers used in calculating the average except the most recent are

```
CYCLE LENGTH = 4
TESt input
  = 100 + STEP(20, 10)
          * SIN(2 * 3.14159 * (Time - 10) / CYCLE LENGTH)
```

a. Changes in Vensim equations from Figure 6.2a



b. Amplitude change and phase shift

Figure 6.3 *Third order delay impact on amplitude and phase*

delayed. The delays in an averaging process can have an important impact on how the process responds to changes in the external environment.

Moving Average

The most obvious averaging procedure is a *moving average.* With this approach, data is used from a specified period (for example, daily sales figures for the last month), and each of the entries used in the average is given the same weight in calculating the average. Thus, the average is calculated by simply adding up the entries and dividing by the number of entries. As time moves along, the average is recalculated by dropping the oldest entry each time period and adding on the most recent entry.

The moving average is a simple procedure to explain and implement. Most people learn how to calculate averages in elementary school, and adding the moving component to the procedure is straightforward.

However, this process has one disadvantage, either as a formal calculation procedure, or as a model for the informal averaging that most people do in their heads every day as they take actions. This disadvantage is that the moving average gives the same weight to all of the entries in the average, regardless of how old they are. Thus, if a moving average of daily sales for the last month is calculated, the sales figure for a month ago receives as much weight in calculating the average as the sales figure for yesterday. However, conditions change, and it seems that often the sales figure for yesterday should receive more weight, at least if the average is going to be used as a basis for taking some action. After all, yesterday s sales were made under conditions that are the closest to today s conditions. Especially in a model of how people informally average information, it seems clear that recent data should receive more weight. When we think about past conditions, we are generally give more weight to what has just happened then to conditions days, weeks, or months ago.

Exponential Smoothing

A straightforward averaging process which gives more weight to recent data is *exponential smoothing.* With this process, each successively older entry used in the average receives proportionately less weight in calculating the average, and the ratio of the weights for each successive pair of data points is the same. Thus, if the ratio of the weights for the most recent and second most recent data points is 0.8, then the ratio of the weights for any entry and the next older entry will also be 0.8. This implies, for example, that the ratio of the weights between the most recent and the third most recent entries will be $0\ 8 \times 0\ 8 = 0\ 64$.

In concept, exponential smoothing considers a set of data stretching back infinitely far into the past. However, the weights for older data points become smaller as new data points are added, and thus older data points have successively less impact on the average as time goes on.

Most simulation systems include functions to implement exponential smoothing. In Vensim, this is done with functions SMOOTH and SMOOTHI. The use of SMOOTH is illustrated in Figure 6.4. SMOOTH takes two arguments, the first of the which is the input to be smoothed, and the second is the averaging constant which is used to set the ratios of the weights for the weighted exponential average. Figure 6.4a shows a stock and flow diagram for an exponential average. The information arrow over which the averaging is taking place is shown bolder than a standard information arrow. (This notation is useful to quickly show links over which averaging is done, but it is not universally used. In some stock and flow diagrams, there is no indication given of a smoothed input.)

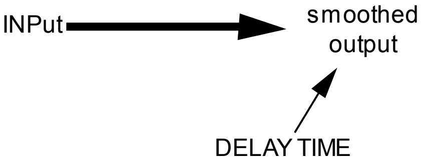
Figure 6.4b shows the Vensim equations for the smoothing operation. Equation 7 shows how the smoothing equation is entered.

Figure 6.4c shows the result of smoothing a step input. As we would expect from the discussion above, the smoothed version moves slowly from the original level of the input to the final level.

The starting value for the SMOOTH output is set equal to the initial value of the input. The function SMOOTHI allows you to set this starting value to a different level if desired.

6.4 Information Delays

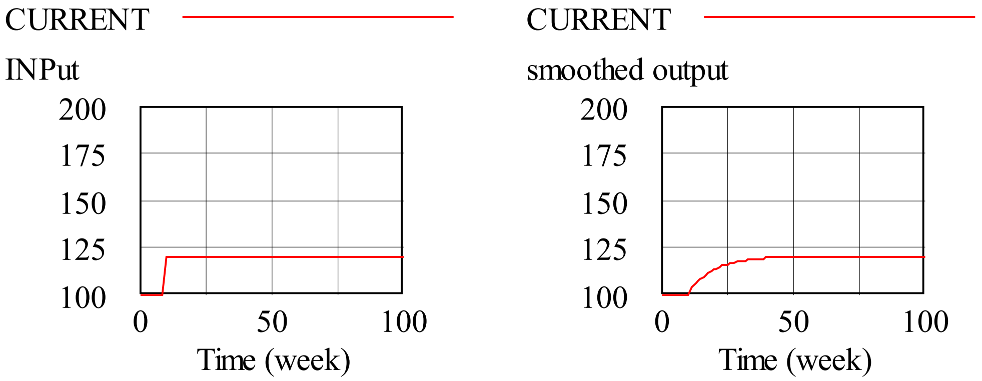
Transmission of information can be delayed in transit in the same way that material flow can be delayed. The various material flow delay functions can be used to model information delay. However, in situations where the length of the delay can vary over time, the behavior of delayed information can differ from the behavior of delayed material. Delayed material must be conserved, even if the amount of the delay varies. On the other hand, some of the delayed information can be forgotten if the delay varies. Vensim provides the functions DELAY INFORMATION, SMOOTH3 AND SMOOTH3I to model this phenomenon. Consult the reference manuals for further information about these functions.



a. Stock and flow diagram

```
(1) DELAY TIME = 10
(2) FINAL TIME  = 100
(3) INITIAL TIME  = 0
(4) INPut = 100 + STEP(20, 10)
(5) SAVEPER  = TIME STEP
(6) smoothed output = SMOOTH(INPut, DELAY TIME)
(7) TIME STEP  = 1
```

b. Vensim equations



c. Smoothing

Figure 6.4 *Exponential smoothing*

CHAPTER 7

Representing Decision Processes

Decisions processes are the glue that binds together the information and material flow networks in an organization. Decisions about what information to collect and how to process it determine how information flows into an information network at the points where this network originates on the material flow network. Similarly, decisions about how to use information and what actions to take on material flows determine how information will impact those material flows at points where the information network points into the material flow network. Thus, a critical aspect of creating useful simulation models is appropriately modeling decision processes.

Increasingly, decision making is automated as computers take over more routine decision making activities in business processes, but many decisions continue to be made by humans. Thus, it is necessary to model human decision making if a realistic model is to be constructed of a process. This may seem like an overwhelmingly complex undertaking. How can we hope to mimic the subtle nuances of the human mind? Surely this is a task beyond the capabilities of a computer model!

This chapter presents research results about human reasoning, and then considers how to model decision making in a simulation model. As we shall see, the research results strongly support the conclusion that human decision making is neither particularly complex nor particularly effective. This somewhat discouraging result does, however, carry an optimistic message for those interested in modeling and improving business processes: It is possible to model human decision making with relatively simple models, and it is also possible to improve on unaided human decision making with systematic decision policies.

7.1 Experts and Expertise

This section summarizes key points in Chapter 10, Proper and Improper Linear Models, of Dawes (1988). That chapter presents results of research on the ability of experts to provide accurate intuitive predictions. The research results strongly support the conclusion that experts are *not* good intuitive predictors and that simple models using the same predictor variables as the experts provide more accurate predictions. Page references with the following quotes refer to Dawes (1988) unless otherwise noted.

Research Findings

A large number of studies have addressed the question of whether trained experts intuitive global predictions are better than statistically derived weighted averages (linear models) of the relevant predictors. Dawes notes (pp. 205 6), This question has been studied extensively by psychologists, educators, and others interested in predicting such outcomes as college success, parole violation, psychiatric diagnosis, physical diagnosis and prognosis, and business success and failure   In 1954 Meehl   summarized approximately twenty such studies comparing the clinical judgment method with the statistical one. *In all studies, the statistical method provided more accurate predictions, or the two methods tied.* Approximately ten years later, Jack Sawyer reviewed forty-five studies comparing clinical and statistical prediction. Again, there was *not a single study* in which clinical global judgment was superior to the statistical prediction.

Continuing, Dawes says (pp. 207 8), The finding that linear combination is superior to global judgment is strong; it has been replicated in diverse contexts, and *no* exception has been discovered   Meehl was able to state thirty years after his seminal book was published, *There is no controversy in social science which shows such a large body of qualitatively diverse studies coming out so uniformly in the same direction as this one*   People have great misplaced confidence in their global judgments.

The referenced work compared statistically derived weighted averages of relevant predictors, however, Dawes (pp. 208 9) himself investigated the possibility that *any* linear model might outperform experts. While, as he notes, the possibility seemed absurd, he found in several studies that linear models where the weights were selected randomly except for sign outperformed the experts and did almost as well as those with statistically derived weights.

Discussion of the Research Findings

When first studying these research findings, they may appear to say that experts can be replaced by simple linear equations (with random weights, no less!). However, closer consideration of the research shows that this is too strong a conclusion to reach from the research. Dawes (1979) notes, The linear model cannot replace the expert in deciding such things as what to look for,   it is precisely this knowledge of what to look for in reaching the decision that is the special expertise people have. [However,] people especially the experts in a field are much better at selecting and coding information than they are at integrating it.

Dawes proposes (pp. 212-215) that the findings can be explained by a principle of nature, a mathematical principle, and a psychological principle. The *principle of nature* that Dawes states is that interactions among predictor variables tend to be monotone in many situations of interest. That is, while there may be interactions among predictor variables, these interactions do not change the monotonicity between a particular variable and the prediction [i.e., more of the predictor variable always predicts more (less) of the predicted variable regardless of the levels of the other variables].

The related *mathematical principle* is that interaction effects among variables which contribute monotonically to the overall effect can often be ignored and the resulting linear model will still provide adequate predictions, and also that specific coefficients for predictor variables are not as important in determining the results of a linear model as the signs of these coefficients. (While Dawes terms this principle *mathematical*, it is really an empirical statistical observation concerning real world data since counterexamples can be constructed.)

The *psychological principle* explaining the superior predictive ability of linear models is that people have difficulty integrating more than one variable. Thus, they tend to anchor on a particular predictor variable while making a prediction and do not adjust their predictions sufficiently to account for other variables. Linear models, of course, give constantly proportional attention to all variables.

Dawes concludes (p. 215), Given that monotone interactions can be well approximated by linear models (a statistical *fact*), it follows that because most interactions that do exist in nature are monotone and because people have difficulty integrating information from noncomparable dimensions, linear model will outperform clinical judgment. The only way to avoid this broad conclusion is to claim that training makes experts superior to other people at integrating information (as opposed, for example, to knowing what information to look for), and there is no evidence for that. There is no evidence that experts *think differently* from others.

He further comments (pp. 215 219), The conclusion that [linear models] outperform global judgments of trained experts is not a popular one with experts, or with people relying on them   Experts have been revered and well paid for years for their it is my opinion that judgments   As James March points out, however, such reverence may serve a *purely social function.* People and organizations have to make decisions, often between alternatives that appear equally good or bad. What better way to justify such decisions than to consult any intuitive expert, and the more money she or he charges, the better

But there is also a structural reason for doubting the inferiority of global judgment   When we construct a linear model in a prediction situation, we know exactly how poorly it predicts. In contrast, our feedback about our global judgments is flawed. Not only do we selectively remember our successes, we often have *no knowledge* of our failures   [For example, considering judgments on accepting or rejecting graduate school applicants], who knows what happens to rejected graduate school applicants? Professors have access only to accepted ones, and if the professors are doing a good job, the accepted ones will do well exonerating the professors judgments

In contrast, the systematic predictions of linear models yield data on just how poorly they predict. For example, in [one] study only 18% of the variance in longevity of Hodgkin s disease patients is predicted by the best linear model , but that is opposed to 0% by the world s foremost authority. Such results bring us abruptly to an unpleasant conclusion: a lot of outcomes about which we care deeply are not very predictable   We *want* to predict outcomes important to us. It is only rational to conclude that if one method (a linear model) does not predict well, something else may do better. What is not rational   is to conclude this something else is intuitive global judgment.

Concluding Comments on the Research Findings

The research discussed above carries an optimistic message for those of us who work on quantitative models. Even simple quantitative models can outperform experts in prediction tasks. However, the research also points out that experts play a key role in developing such models: They are needed to identify the key variables to incorporate into a model.

The research also carries a cautionary message for those working on developing computer-based expert systems. The generally stated criterion for judging the effectiveness of such systems is how well they replicate the performance of an expert. However, the research indicates that at least in prediction tasks it is possible with even simple models to outperform experts *once the key predictor variables have been identified*. Thus, the performance of experts may not be a good benchmark for judging the performance of a computer-based expert system. It is probably possible to do better.

As a final comment, I return to the quote from Dawes at the end of the last subsection. While he notes the superiority of linear models over expert judgment, he also notes that these models don t do a particularly good job either in many situations. Often, better models are available, especially when physical system performance is of interest. For example, someone predicting the behavior of a new airplane that has not yet been built would not use either expert judgment or a simple linear model. The physical principles which govern the behavior of an airplane are well known, and a detailed quantitative model would be built to predict the performance of the airplane long before it was built.

7.2 Modeling Decision Processes

The remainder of this chapter presents structures that can be used to represent decision making processes within a simulation model. Specifically, we consider decisions at points where information arrows enter flows. These junctions are key decision making points within an organizational process because they are where information impacts the physical activities of the process.

Example: Managing Flows Through the Thurabond Dam

We will proceed by considering a management decision process with a simple structure: Managing the outflow from a water reservoir. While it is not likely that most readers need to manage the outflow from a reservoir, this decision situation has two characteristics that make it a useful example for models of decision processes. First, the stock and flow variables are graphically obvious: The amount of water in the reservoir is clearly a stock, and the flows into and out of the reservoir are clearly flows. Second, it has a relatively simple structure where the implications of different decision rules can be easily seen. In many business settings, there are several interacting stocks and flows, and thus the impact of changing a single decision rule may be obscured by the complexities of the situation. This example is inspired by one in Roberts, et al (1983), Chapter 22.

The Rappanno Valley has ideal growing conditions for several different types of vegetables, but very little rain. During the waning years of Senator Thurabond s distinguished Congressional career, Federal funds were allocated for the construction of the Thurabond Dam in Big Stormy Gorge on the Callahali River. This dam, together with the Rappanno Valley Irrigation Project, established an extensive irrigation system throughout Rappanno Valley, and in the forty years since the completion of the dam and irrigation system, a prosperous agricultural community has developed there.

The essential features of the reservoir and irrigation system are shown in Figure 7.1. The inflow to Big Stormy Reservoir behind Thurabond Dam is not under our control, and the amount of water in the reservoir is labeled Reservoir Contents. All releases from the reservoir flow into the Rappanno Valley drainage basin where the water is primarily used for agricultural purposes. The amount of water available in the drainage basin at any time for agricultural purposes is labeled Drainage Basin Contents. Water is consumed from the Rappanno Valley drainage basin in a variety of ways, including transpiration from plants, evaporation, and drainage. For notational simplicity, we refer to all of these losses as drainage. This drainage is not under the control of the Thurabond Dam operator. Thus, there is only one decision variable, the release through Thurabond Dam, which is shown in the center of Figure 7.1.

We will examine decision policies for managing releases through Thurabond Dam for use in Rappanno Valley agriculture. The Dam impounds water from an substantial stretch of the Callahali River, and the average net annual impoundment, taking into account evaporation losses, is 0.5 million acre-feet. Standard

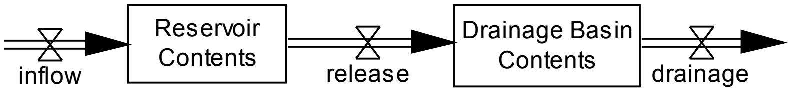


Figure 7.1 *Water flow process*

operating procedure at Thurabond Dam is to maintain a long term average of one million acre-feet of water behind the dam in Big Stormy Reservoir, although the actual amount of water in the reservoir may vary over the short term depending on rainfall and other conditions. Not surprisingly, agriculture has expanded in the Rappanno Valley to consume 0.5 million acre-feet per year of water. More specifically, the drainage system within the Valley holds one million acres-feet of water accessible for agricultural use, and fifty percent of this is consumed each year.

A reservoir system can have several purposes. The rainy season in a region may not coincide with the growing season, and then a reservoir can be used to time shift water from the rainy season to the growing season. If there is flooding in an area, then the reservoir can trap water during periods of high flow and gradually release it over an extended period of time. If there are periods of drought, then the reservoir can save water over several years and release it during drought years.

The primary purpose of Big Stormy Reservoir is to hold water that would otherwise flow unused down the Callahali River to the ocean about two hundred miles away, so that this water can be used for agriculture. Our analysis of operating policies for releases through Thurabond Dam will focus on maintaining sufficient flow to support agriculture in the Rappanno Valley, while assuring that there is sufficient reserve in Big Stormy Reservoir to continue to support agriculture through a drought period, and also assuring that water does not build up in the reservoir to the point where it threatens to overtop Thurabond Dam.

It is important to note that in this situation, as in any process involving flow of material, the flowing material must be *conserved.* That, is the total amount of material that flows into the process must, over the long run, average out to the same amount that flows out of the process. Otherwise, material will indefinitely continue to pile up somewhere in the process and you will ultimately run out of storage capacity. Since there is a sequential flow of water through the process shown in Figure 7.1, the requirement that material must be conserved means that the long run averages for inflow, release, and drainage must all be the same.

Types of Decision Models

The discussion about experts earlier in this chapter shows that even experts in a field use relatively simple decision procedures. Therefore, it is often appropriate to use simple models to represent decision processes in simulation models. There are two primary issues that must be addressed in constructing such models: 1) What factors should be taken into account in the decision model, and 2) How should these factors be combined. We investigate both of these issues below.

Many decision processes take into account multiple factors. For Thurabond Dam, it seems clear that any reasonable decision process for releases will need to consider both the level of Big Stormy Reservoir and the impact of outflows from the reservoir on agriculture in Rappanno Valley. In this case, and this is typical of many decision situations, there are explicit or implicit *goals* with regard to both of these factors. For the reservoir level: We do not want the quantity of water in the reservoir to become so large that a sudden increase in inflow might lead to the threat of overtopping Thurabond Dam. (In such situations, emergency releases must be made, which can lead to substantial downstream flooding.) We also do not want the water in the reservoir to get too low, because if a drought occurs when the reservoir is low we might not be able to provide sufficient water to the Rappanno Valley to support agriculture.

With regard to outflows: We do not want these to be too high or they will cause flooding in the Rappanno Valley, and we also do not want them to be too low because this could cause crop failure. One way to address this is to attempt to maintain a constant value for the contents of the drainage basis. If this gets too high, then flooding will occur, and if it gets too low, there will be insufficient water to maintain crops.

Thus, to summarize, our goals are to maintain constant levels for the two variables Reservoir Contents and Drainage Basin Contents in Figure 7.1.

One can visualize a variety of different quantitative forms for decision functions which address multiple goals. The two simplest are 1) an average of the factors, perhaps with different weights used for each factor, or 2) a product of the factors. These forms are both used in simulation models, and they have proved sufficient to model a variety of different real-world decision processes.

7.3 Weighted-average Decision Models

The ideas underlying a weighted-average decision model for a flow variable are straightforward and intuitively appealing:

1. A portion of the flow is being used to attempt to maintain some goal with respect to each of the decision factors, and if the flow deviates from what is needed to maintain that goal, then this portion of the flow should be adjusted.
2. These adjustments are made over a period of time (that is, averaged) in order to avoid disruptive discontinuities in operations, and also to smooth out transient shifts in conditions due to random factors.
3. The total flow is made up of a sum of the portions assigned to achieving each goal.

4 Different weights may be assigned to meeting each goal depending on their relative importance.

Figure 7.2a shows a stock and flow diagram to represent a weighted-average decision model for the release decision variable. This has been developed from the Figure 7.1 diagram by adding a variety of auxiliary variable, most of which are related to the release decision. In the upper left corner of the diagram, LONG TERM AVERAGE INFLOW is a constant which provides the average flow rate into Big Stormy Reservoir. From our earlier discussion, we know that this is 0.5 million acre-feet per year. This is used to set a target for the amount of water in Big Stormy Reservoir, which is indicated on the diagram by reservoir target. We will assume that the target is two times the LONG TERM AVERAGE INFLOW. That is, the reservoir is operated to maintain on average two years of inflow.

There is also a target for the amount of accessible water in the Rappanno Valley drainage system, which is indicated in the upper right corner of Figure 7.2a by DRAINAGE BASIN TARGET. This target is set to maintain a constant amount of water in the basin over the long term. As noted above, annual drainage from the valley is fifty percent of the accessible water in the basin. We also know from our earlier discussion that this average drainage must equal the LONG TERM AVERAGE INFLOW, which is 0.5 million acre-feet. Therefore, the DRAINAGE BASIN TARGET must be twice this, or one million acre-feet.

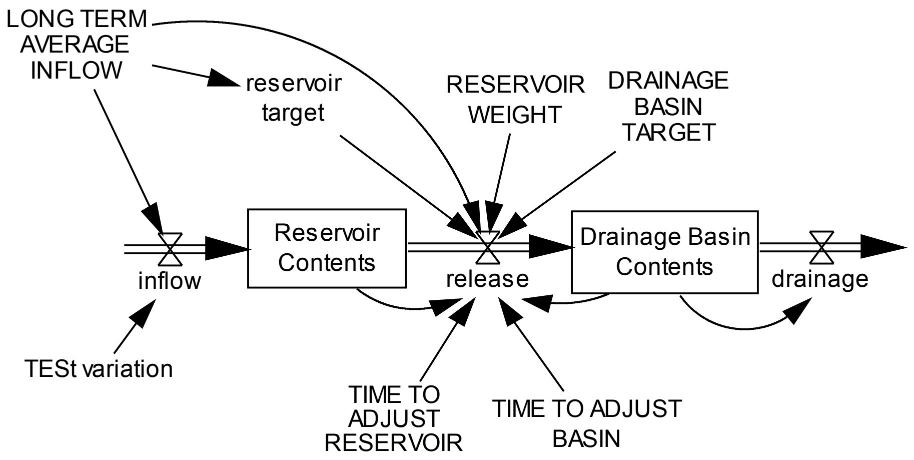
The two constants TIME TO ADJUST RESERVOIR and TIME TO ADJUST BASIN, which are shown in the lower center of Figure 7.2a, relate to the averaging period used to address deviations from the goals with respect to the reservoir contents and the drainage basin contents. Finally, the constant RESERVOIR WEIGHT in the upper center of the figure represents the weight assigned to the goal of maintaining a constant value for Reservoir Constant, relative to maintaining a constant value for Drainage Basin Contents.

As noted above, with a weighted-average decision rule, the flow is visualized as being split into several parts which add up to constitute the entire flow. A useful way to develop the decision rule is often to visualize the flow being controlled as made up of a base component needed to maintain stable conditions over the long run, and then correction terms needed to address deviations from each of the goals. For the reservoir, the long term average inflow to the reservoir is LONG TERM AVERAGE INFLOW, and therefore the base component of release must be equal to this.

The correction term for deviations from the target for the quantity of water in the reservoir can then be built up in three steps: First, note that this correction term should be zero when the value of Reservoir Contents is equal to reservoir target. Therefore, the correction term should be proportional to

$$\text{Reservoir Contents} - \text{reservoir target} \qquad (7\ 1)$$

That is, if there is more water in the reservoir than the target, then the release should be increased, while if there is less water than the target, then release should be decreased.



a. Stock and flow diagram

```
(01) drainage = 0.5 * Drainage Basin Contents
(02) Drainage Basin Contents = INTEG(release-drainage,
       DRAINAGE BASIN TARGET)
(03) DRAINAGE BASIN TARGET = 1
(04) FINAL TIME  = 4
(05) inflow = LONG TERM AVERAGE INFLOW+TEST variation
(06) INITIAL TIME  = 0
(07) LONG TERM AVERAGE INFLOW = 0.5
(08) release = LONG TERM AVERAGE INFLOW
      + RESERVOIR WEIGHT * (Reservoir Contents - reservoir target)
        / TIME TO ADJUST RESERVOIR
      +(1 - RESERVOIR WEIGHT)
        * (DRAINAGE BASIN TARGET - Drainage Basin Contents)
          / TIME TO ADJUST BASIN
(09) Reservoir Contents
      = INTEG(inflow - release, reservoir target)
(10) reservoir target = 2 * LONG TERM AVERAGE INFLOW
(11) RESERVOIR WEIGHT = 0.5
(12) SAVEPER  = TIME STEP
(13) TESt variation = STEP(0.1, 0.5)
(14) TIME STEP  = 0.01
(15) TIME TO ADJUST BASIN = 0.05
(16) TIME TO ADJUST RESERVOIR = 0.5
```

b. Vensim equations

Figure 7.2 *Weighted-average decision rule*

However, if the expression in equation 7.1 were used as the correction term for deviations from the reservoir goal, this would mean that any deviation would be instantly corrected. This is probably not feasible due to physical constraints on the dam, and it may also not be desirable because every little random variation in reservoir level would result in fluctuations in the release. Thus, the correction will be averaged over a period of time as follows:

$$\frac{\text{Reservoir Contents} - \text{reservoir target}}{\text{TIME TO ADJUST RESERVOIR}} \tag{7 2}$$

This means that if the correction were to continue at the same rate, it would take a length of time equal to TIME TO ADJUST RESERVOIR to completely remove the deviation. (In actuality, the level of the reservoir will change over time, and thus the actual correction period will probably differ from TIME TO ADJUST RESERVOIR.)

Another way to visualize this is to define

$$\text{RESERVOIR ADJUSTMENT FACTOR} = \frac{1}{\text{TIME TO ADJUST RESERVOIR}}$$

and then equation 7.2 can be rewritten

$$\begin{aligned}&\text{RESERVOIR ADJUSTMENT FACTOR}\\ &\qquad\qquad \times (\text{Reservoir Contents} - \text{reservoir target})\end{aligned}$$

From this, we see that RESERVOIR ADJUSTMENT FACTOR is the portion of the deviation from the goal that is corrected each unit of time.

Finally, to complete the correction factor for the reservoir goal, the expression in equation 7.2 is multiplied by the RESERVOIR WEIGHT, which is a number between zero and one, to take into account the relative importance of this goal. This gives

$$\text{RESERVOIR WEIGHT} \times \frac{\text{Reservoir Contents} - \text{reservoir target}}{\text{TIME TO ADJUST RESERVOIR}} \tag{7 3}$$

A similar procedure can be used to determine the correction factor for the drainage basin goal, which is

$$\begin{aligned}&(1 - \text{RESERVOIR WEIGHT})\\ &\qquad \times \frac{\text{DRAINAGE BASIN TARGET} - \text{Drainage Basin Contents}}{\text{TIME TO ADJUST BASIN}}\end{aligned} \tag{7 4}$$

Note that in this case, the actual level for the variable (Drainage Basin Contents) is subtracted from the goal (DRAINAGE BASIN TARGET) because we wish to decrease the flow if the actual level is above the target and increase it is the actual level is below the target. Note also that we are assigning weights to the two goals so that these weights add up to one. Therefore, it is not necessary to define a separate weight for the drainage basin goal: It must be equal to 1 – RESERVOIR WEIGHT.

The final complete expression for the release decision rule is obtained by adding the two correction terms in equations 7.3 and 7.4 to the long term average flow rate LONG TERM AVERAGE INFLOW. This yields

$$\begin{aligned}\text{release} = {} & \text{LONG TERM AVERAGE INFLOW} \\ & + \text{RESERVOIR WEIGHT} \times \frac{\text{Reservoir Contents} - \text{reservoir target}}{\text{TIME TO ADJUST RESERVOIR}} \\ & + (1 - \text{RESERVOIR WEIGHT}) \\ & \quad \times \frac{\text{DRAINAGE BASIN TARGET} - \text{Drainage Basin Contents}}{\text{TIME TO ADJUST BASIN}}\end{aligned} \tag{7 5}$$

The complete set of equations for the reservoir management model with a weighted-additive decision rule are given in Figure 7.2b. Equation 8 of this figure corresponds to equation 7.5 above. The values assumed for the various constants are also shown in Figure 7.2b. Note, also that equation 1 in this figure shows that the drainage is a proportion of the Drainage Basin Contents as discussed above.

As shown by equations 2 and 9 in Figure 7.2b, the initial values of Reservoir Contents and Drainage Basin Contents are set equal to the targets for these variables. Thus, so long as inflow continues to be equal to LONG TERM AVERAGE INFLOW, the entire process will be in steady state, and the levels of the two stocks will remain the same with a constant flow of LONG TERM AVERAGE INFLOW through the system. Referring back to Figure 7.2a for a moment, note that there are dashed arrows from reservoir target to Reservoir Contents, and also from DRAINAGE BASIN TARGET to Drainage Basin Contents. These dashed arrows indicate that the initial level for each of the levels depends on the specified variable, as shown by equations 2 and 9 of Figure 7.2b.

As a test input for this simulation model, we use a step, as shown by equations 5 and 13 of Figure 7.2b. The results are shown in Figure 7.3. The top set of graphs shows the results with a RESERVOIR WEIGHT equal to one, the middle set of graphs shows the results with a RESERVOIR WEIGHT equal to 0.5, and the bottom set of graphs shows the results with a RESERVOIR WEIGHT equal to zero. Thus, in the top and bottom sets of graphs, only one of the goals is taken into account in setting the reservoir release, while in the middle set of graphs both goals are taken into account. (Note that some scales on corresponding graphs in the three parts of Figure 7.3 differ.)

The pattern for release is substantially different for the three cases. When there is no weight on the reservoir goal (RUN0) the release remains constant at 0.5 million acre-feet per year, and the Reservoir Contents steadily grows to absorb the extra inflow that is not being released. When there is no weight on the drainage basin goal (RUN10) the release grows to 0.6 million acre feet per year to stabilize the amount of water in the reservoir, but the Drainage Basin Contents grows substantially.

Finally, in the case where the two goals are given equal weight (RUN5), the results are intermediate between the other two cases, but the values for Reservoir Contents and Drainage Basin Contents are closer to the RUN0 case than the RUN10 cases. The reason for this can be seen from examining the values for the two constants TIME TO ADJUST BASIN and TIME TO ADJUST RESERVOIR in equations 15 and 16 of Figure 7.2b. We see from these equations that TIME TO ADJUST BASIN is one-tenth of TIME TO ADJUST RESERVOIR (0.05 versus 0.5). Thus, adjustments to Drainage Basin Contents are made much more quickly than adjustments to Reservoir Contents, and hence the final results for the equal weight case are closer to the case where all the weight is placed on maintaining a constant value for Drainage Basin Contents. This illustrates that the overall performance of a weighted-average decision rule is equally impacted by the weights and the adjustment time constants.

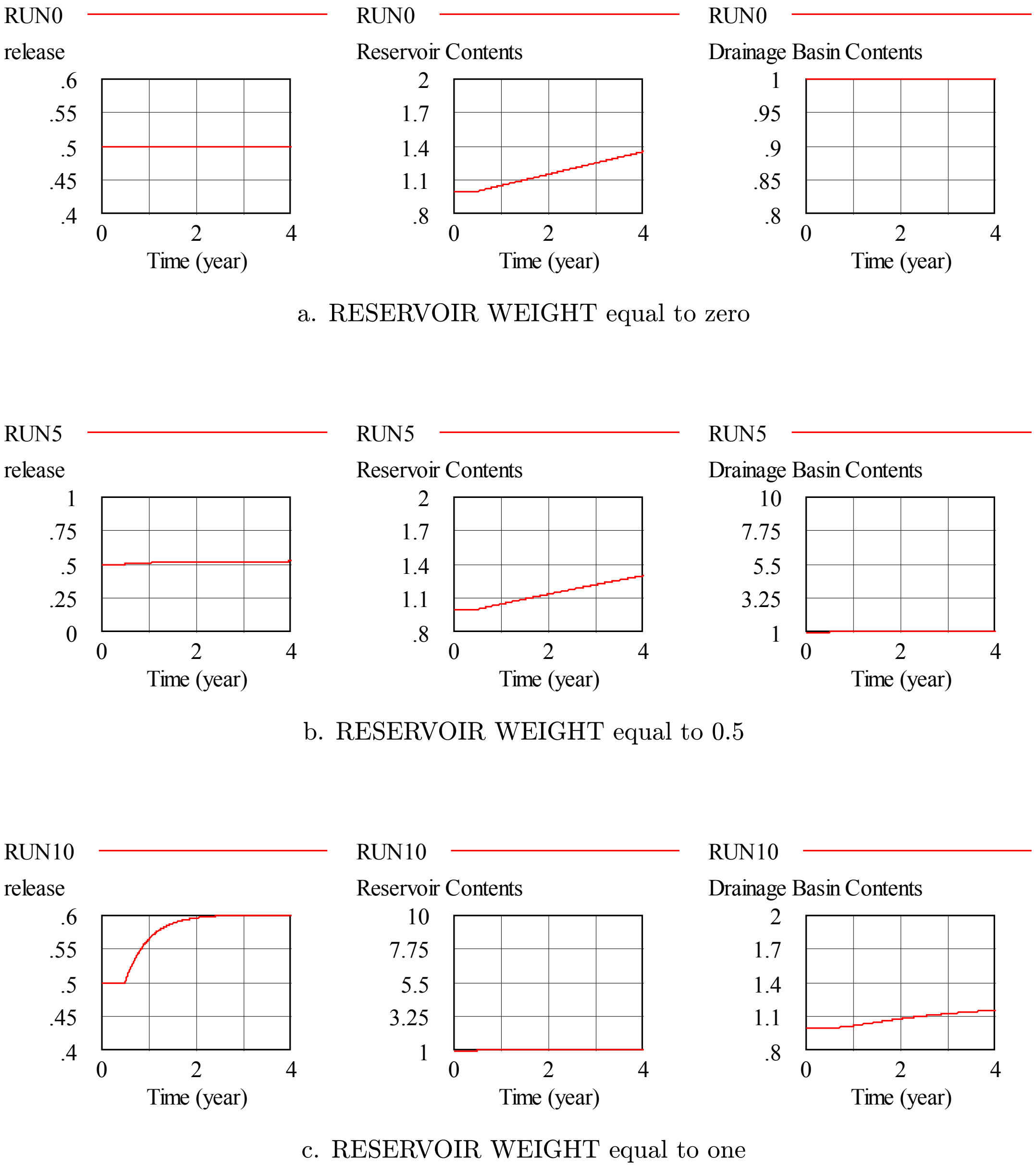
7.4 Floating Goals

The decision model discussed in the last section assumes that long term averages are known and are stable, and therefore can be used in decision rules. What if these long term averages are not known? The model in Figure 7.4 shows one way to address this. The differences between the stock and flow diagram in Figure 7.4a and the one shown earlier in Figure 7.2a are in the upper left hand corner. A new variable short term average inflow has been introduced, and now this is used as an input to the release decision, rather than LONG TERM AVERAGE INFLOW.

This short term average inflow is calculated by smoothing inflow using a first order exponential smooth with a time constant of INFLOW AVERAGING TIME. Thus, this approach does not assume that the decision maker managing release has access to the values of LONG TERM AVERAGE INFLOW. This modeling approach is sometimes called floating goals because the goal is calculated from data generated as the model solves rather than being prespecified. Therefore, this goal can vary, or float as the data changes.

The equations for this model are shown in Figure 7.4b. These differ from the equations in Figure 7.2a as follows: Additional equations numbered 6 and 14 have been added to calculate short term average inflow.

The results of applying the floating goal decision rule are shown in Figure 7.5. The meanings of RUN0, RUN5, and RUN10 are the same in this figure as in Figure 7.3. Note that in this case when all the weight is put on the reservoir goal (RUN10) there is a somewhat counterintuitive result with respect to release. After the inflow to the reservoir jumps at time 0.5, the release actually drops for about a year. This is because the target for Reservoir Contents grows as the inflow to the reservoir grows, as shown by equation 11 of Figure 7.4b, and therefore more water is needed in the reservoir to meet the target.

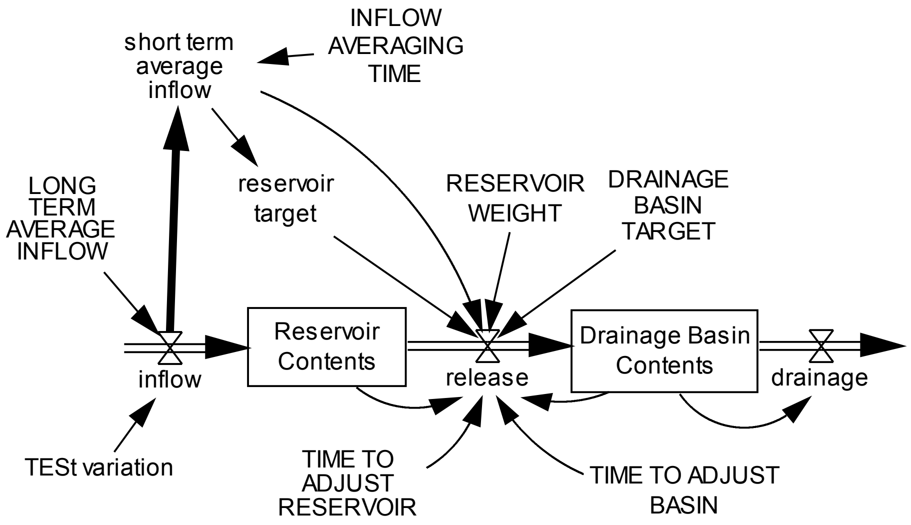


a. RESERVOIR WEIGHT equal to zero

b. RESERVOIR WEIGHT equal to 0.5

c. RESERVOIR WEIGHT equal to one

Figure 7.3 *Dynamics with weighted-average decision rule*

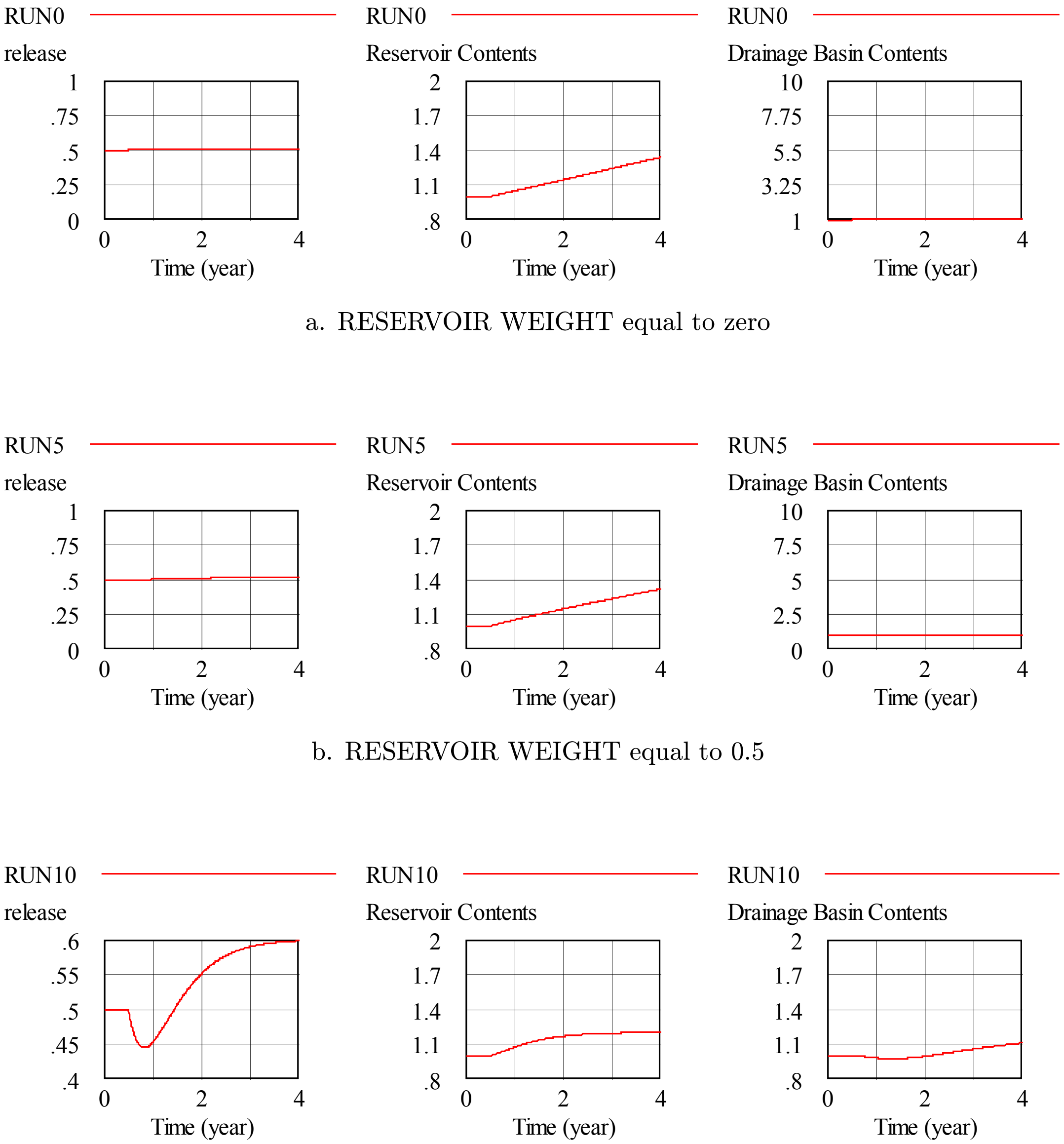


a. Stock and flow diagram

```
(01) drainage = 0.5 * Drainage Basin Contents
(02) Drainage Basin Contents
        = INTEG(release - drainage, DRAINAGE BASIN TARGET)
(03) DRAINAGE BASIN TARGET = 1
(04) FINAL TIME  = 4
(05) inflow = LONG TERM AVERAGE INFLOW + TESt variation
(06) INFLOW AVERAGING TIME = 0.5
(07) INITIAL TIME  = 0
(08) LONG TERM AVERAGE INFLOW = 0.5
(09) release = short term average inflow +
      RESERVOIR WEIGHT
         * (Reservoir Contents - reservoir target)
             / TIME TO ADJUST RESERVOIR
     +(1 - RESERVOIR WEIGHT)
         * (DRAINAGE BASIN TARGET - Drainage Basin Contents)
          /TIME TO ADJUST BASIN
(10) Reservoir Contents = INTEG(inflow - release, reservoir target)
(11) reservoir target = 2 * short term average inflow
(12) RESERVOIR WEIGHT = 0.5
(13) SAVEPER  = TIME STEP
(14) short term average inflow = smooth(inflow, INFLOW AVERAGING TIME)
(15) TESt variation = STEP(0.1, 0.5)
(16) TIME STEP  = 0.01
(17) TIME TO ADJUST BASIN = 0.05
(18) TIME TO ADJUST RESERVOIR = 0.5
```

b. Vensim equations

Figure 7.4 *Floating goal decision rule*

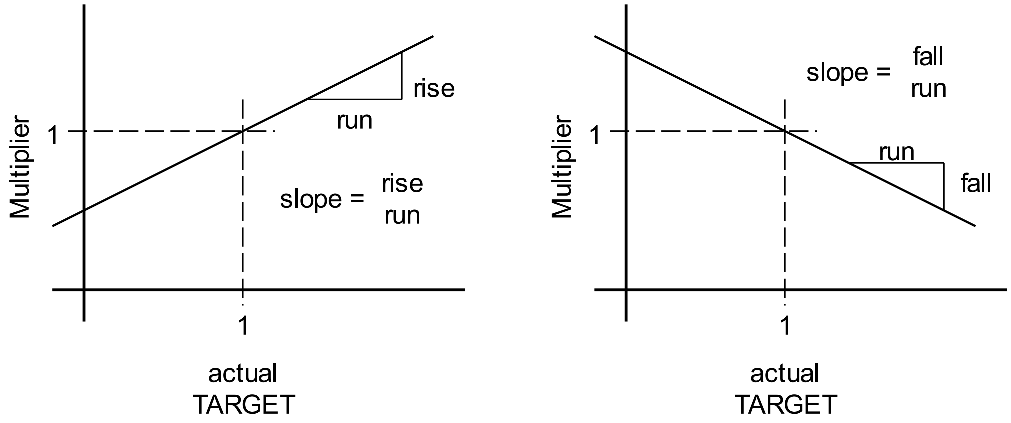


a. RESERVOIR WEIGHT equal to zero

b. RESERVOIR WEIGHT equal to 0.5

c. RESERVOIR WEIGHT equal to one

Figure 7.5 *Dynamics with floating goal decision rule*



a. Increasing case

b. Decreasing case

Figure 7.6 *Multipliers*

7.5 Multiplicative Decision Rule

Another approach for modeling decision rules is to use a multiplicative form. With the weighted-additive form, correction terms are *added* to a base flow rate, while with the multiplicative form, correction factors are used to *multiply* the base flow rate. The correction factors are illustrated in Figure 7.6. The left hand graph in this figure applies to a situation where if the variable of interest is above its target (goal) value the flow needs to be increased. (This is the situation in the reservoir example for Reservoir Contents.) The right hand graph in Figure 7.6 applies to a situation where if the variable of interest is above its target value, the flow needs to be reduced. (This is the situation in the reservoir example for Drainage Basin Contents.)

It turns out to be useful to normalize the variables by dividing them by their target values, as shown in the Figure 7.6 graphs. When this is done, a situation where a normalized variable is equal to one will have a multiplier of one. That is, when the value of a variable is equal to its target value, there will be no correction applied to the base case flow.

The slope of the normalized curve, as defined in the graphs in Figure 7.6, then sets the strength of the reaction in the flow that occurs for a specified percentage deviation in a variable from its target value. The greater the slope, the greater the response for a specified percentage deviation of a variable.

It is straightforward to derive the equation for the multiplier as a function of the variable, its TARGET, and its slope. For the increasing case in Figure 7.6a, this is

$$\text{Multiplier} = \text{slope} \times \frac{\text{actual}}{\text{TARGET}} + (1 - \text{slope}) \qquad (7\ 6a)$$

and for the decreasing case in Figure 7.6b, this is

$$\text{Multiplier} = 1 + \text{slope} - \text{slope} \times \frac{\text{actual}}{\text{TARGET}} \qquad (7\ 6b)$$

The results of applying the multiplicative decision rule approach to the reservoir example are shown in Figure 7.7. (Note that this example is the multiplicative version of the weighted-average example in Figure 7.2. If desired, a floating goals approach can be applied to the multiplicative case in a manner analogous to that presented above for the weighted-average decision rule.) As in the weighted-average case, the base case flow is LONG TERM AVERAGE INFLOW. The targets for the reservoir and drainage basin are reservoir target and DRAINAGE BASIN TARGET, respectively. The slopes of the correction factors for these are RESERVOIR ADJUSTMENT SLOPE and DRAINAGE BASIN ADJUSTMENT SLOPE, respectively.

The stock and flow diagram for this decision rule is shown in Figure 7.7a, and the Vensim equations are shown in Figure 7.7b. Equation 9 of this figure shows how equation 7.6 is applied in this case to develop the multiplicative decision rule.

Figure 7.8 shows the results of running a simulation with the equations in Figure 7.7b.

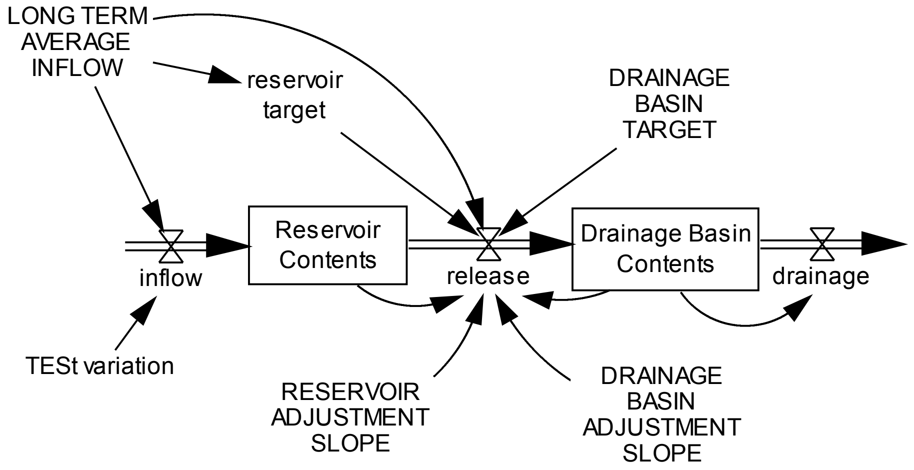
The performance of the weighted-average and multiplicative decision rules will be similar for small variations from the desired flow, provided the constants in the two models are suitably adjusted. For larger variations, the multiplicative rule can lead to a more aggressive response than the weighted-average rule because the responses for the two variables interact in a multiplicative fashion. This type of decision rule may be appropriate for modeling some decision makers. However, the discussion above of the performance of experts indicates that an additive model will perform as well as many actual decision makers.

7.6 References

Dawes, R. M. 1979. The Robust Beauty of Improper Linear Models in Decision Making. *American Psychologist* **34**, 571 582.

Dawes, R. M. 1988. *Rational Choice in an Uncertain World*. Harcourt Brace Jovanovich, San Diego.

N. Roberts, D. F. Anderson, R. M. Deal, M. S. Garet, and W. A. Shaffer, *Introduction to Computer Simulation: The System Dynamics Approach*, Addison-Wesley, Reading, MA, 1983.



a. Stock and flow diagram

```
(01) drainage = 0.5*Drainage Basin Contents
(02) DRAINAGE BASIN ADJUSTMENT SLOPE = 1
(03) Drainage Basin Contents = INTEG(release-drainage,
       DRAINAGE BASIN TARGET )
(04) DRAINAGE BASIN TARGET = 1
(05) FINAL TIME  = 4
(06) inflow = LONG TERM AVERAGE INFLOW+TEST variation
(07) INITIAL TIME  = 0
(08) LONG TERM AVERAGE INFLOW = 0.5
(09) release = LONG TERM AVERAGE INFLOW
       *(RESERVOIR ADJUSTMENT SLOPE
         *(Reservoir Contents/reservoir target)
           +(1-RESERVOIR ADJUSTMENT SLOPE))
       *(1+DRAINAGE BASIN ADJUSTMENT SLOPE
         -DRAINAGE BASIN ADJUSTMENT SLOPE
           *(Drainage Basin Contents/DRAINAGE BASIN TARGET))
(10) RESERVOIR ADJUSTMENT SLOPE = 1
(11) Reservoir Contents = INTEG(inflow-release,reservoir target)
(12) reservoir target = 2*LONG TERM AVERAGE INFLOW
(13) SAVEPER  =
         TIME STEP
(14) TESt variation = step(0.1,0.5)
(15) TIME STEP  = 0.01
```

b. Vensim equations

Figure 7.7 *Multiplicative decision rule*

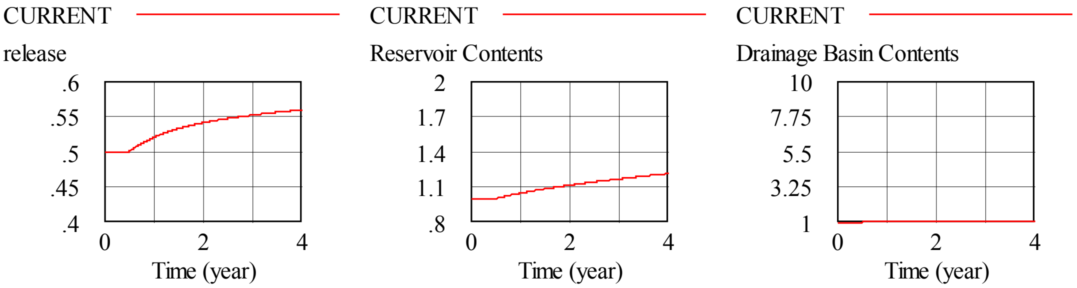


Figure 7.8 *Dynamics with multiplicative decision rule*

CHAPTER 8

Nonlinearities

A process is said to be *linear* if the process response is proportional to the stimulus given to it. For example, if you double the amount deposited in a conventional savings account (the stimulus), then you will receive double the interest (the response). Similarly, if you work ten percent longer hours, you would hope to accomplish ten percent more work. These are linear responses.

Models that assume a process is linear have been extensively studied because the mathematics for such models is relatively straightforward, and linear models can adequately represent the behavior of many realistic processes over a useful range of conditions. It is often possible to solve the equations for linear models without the need to use computers. Thus, in the era before the widespread availability of computers, the ease of solution for linear models led to their use even in situation where the real-world process was known to be *nonlinear*.

Many business processes are nonlinear, especially when pressed to extremes. For example, while it may be true that if you work ten percent longer hours you will accomplish ten percent more work, it is probably not true that if you work twice as many hours you will accomplish twice as much work. Many of us have attempted to do this, and have soon suffered from burnout leading to a reduction in our working effectiveness. This is a nonlinear response. Similarly, the available production capacity may limit the amount of a product that can be sold, regardless of the amount of sales effort or the degree of customer demand.

In other cases, such as graduated income taxes or variable interest rates on money market accounts, nonlinear responses are deliberately designed into the system. With graduated income taxes, the amount of tax grows more rapidly than the increase in income, and with a money market account the rate of interest may grow more than proportionally as the balance grows.

The simulation approach presented in preceding chapters can be extended to address nonlinear effects without much difficulty. This capability for readily modeling nonlinear processes is an advantage of simulation over hand calculation methods. With hand calculation, nonlinear situations can be complex to address. With simulation, it is often as straightforward to model nonlinear situations as to model ones that are linear.

8.1 Nonlinear Responses

Figure 8.1 shows the stock and flow diagram, Vensim equations, and a graph of Savings Balance for a conventional savings account with compound interest where the interest is left to accumulate in the account for 20 years. The interest rate is five percent (0.05) per year, and the initial balance is $900. After 20 years, the balance has grown to a little over $2,400. The response (that is, the earned interest) is linearly related to the initial amount placed in the account.

Using IF THEN ELSE to Model Nonlinear Responses

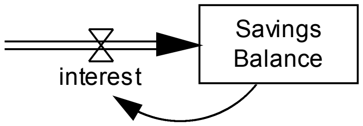
Some money market accounts have a sliding interest rate where the interest rate depends on the balance in the account. For example, suppose that an interest rate of five percent (0.05) per year is paid on every dollar in the account up to $1,000, and an interest rate of ten percent (0.10) per year is paid on every dollar in the account over $1,000. Then the interest is given by

$$\text{interest} = \begin{array}{ll} 0\ 05 \times \text{Savings Balance} & \text{Savings Balance} < \$1\ 000 \\ 0\ 05 \times 1\ 000 & \\ \quad +0\ 10 \times (\text{Savings Balance} - 1\ 000) & \text{otherwise} \end{array}$$

A somewhat generalized version of this model is shown in Figure 8.2. In Figure 8.2, the Savings Balance amount at which the interest rate changes is specified by the constant BREAKPOINT (which is 1,000 for this example), the interest rate paid on each dollar below BREAKPOINT is specified by the constant LOW RATE (which is 0.05), and the interest rate paid on each dollar above BREAKPOINT is specified by the constant HIGH RATE (which is 0.10). (The use of these constants, rather than hard wiring in specific values for BREAKPOINT, LOW RATE, and HIGH RATE, facilitates sensitivity analysis using the automated features of Vensim. This is discussed further below.)

From Figure 8.2c, we see that the Savings Balance after 20 years is over $3,400, which is substantially more than with the conventional savings account shown in Figure 8.1. (Note that the increase in savings rate for most real-world money market savings accounts above the BREAKPOINT is usually not as great as shown in this example! The large value used in this example for HIGH RATE makes it easier to see the impact of the nonlinear interest rate.) A detailed examination of the model output shows that the Savings Balance exceeds the BREAKPOINT value of $1,000 during the second year, and after that the money market account generates more interest than the conventional savings account.

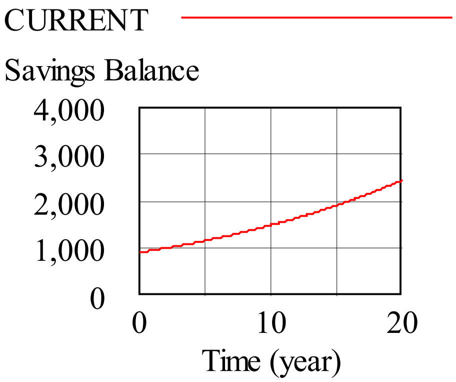
It is straightforward to modify the model in Figure 8.2 to demonstrate that this process is nonlinear. Specifically, the amount of interest earned by the account is not linearly proportional to the initial Savings Balance. As an example, you may wish to verify that when the initial Savings Balance is doubled to $1,800, the final Savings Balance after twenty years almost triples from about $3,500 to almost $10,000. This happens because the modified initial balance of $1,800 is greater than $1,000. Therefore, each dollar of interest earned is compounded at



a. Stock and flow diagram

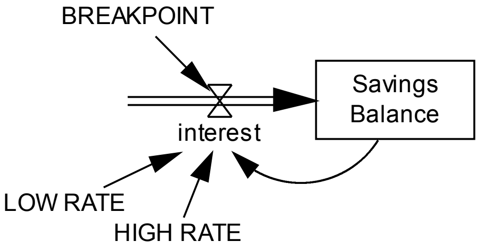
```
(1) FINAL TIME  = 20
(2) INITIAL TIME  = 0
(3) interest = 0.05 * Savings Balance
(4) SAVEPER  = TIME STEP
(5) Savings Balance= INTEG (interest, 900)
(6) TIME STEP  = 0.125
```

b. Vensim equations



c. Savings balance

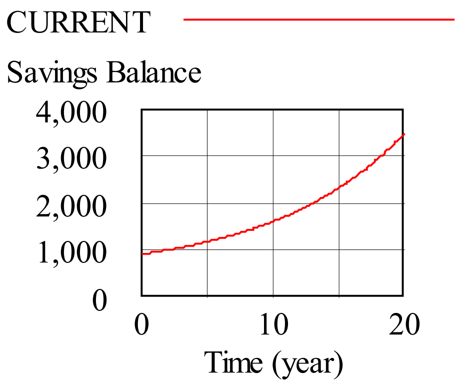
Figure 8.1 *Model for a conventional savings account*



a. Stock and flow diagram

```
(01) BREAKPOINT = 1000
(02) FINAL TIME  = 20
(03) HIGH RATE = 0.1
(04) INITIAL TIME  = 0
(05) interest=
        IF THEN ELSE(Savings Balance < BREAKPOINT,
          LOW RATE * Savings Balance,
          LOW RATE * BREAKPOINT
            + HIGH RATE * (Savings Balance - BREAKPOINT))
(06) LOW RATE = 0.05
(07) SAVEPER  = TIME STEP
(08) Savings Balance= INTEG (interest, 900)
(09) TIME STEP  = 0.125
```

b. Vensim equations



c. Savings balance

Figure 8.2 *Model for a money market savings account (IF THEN ELSE)*

HIGH RATE from the beginning, while this does not happen with the interest for the Initial Balance of $900 specified in Figure 8.2 until the Savings Balance reaches $1,000.

The IF THEN ELSE feature illustrated in equation (05) of Figure 8.2b provides a powerful and flexible way to model this type of nonlinear response. For example, it is possible to nest a second IF THEN ELSE within the first one to handle a situation where there is a second breakpoint at which the interest rate earned on each dollar changes again.

Using Lookup Functions to Model Nonlinear Responses

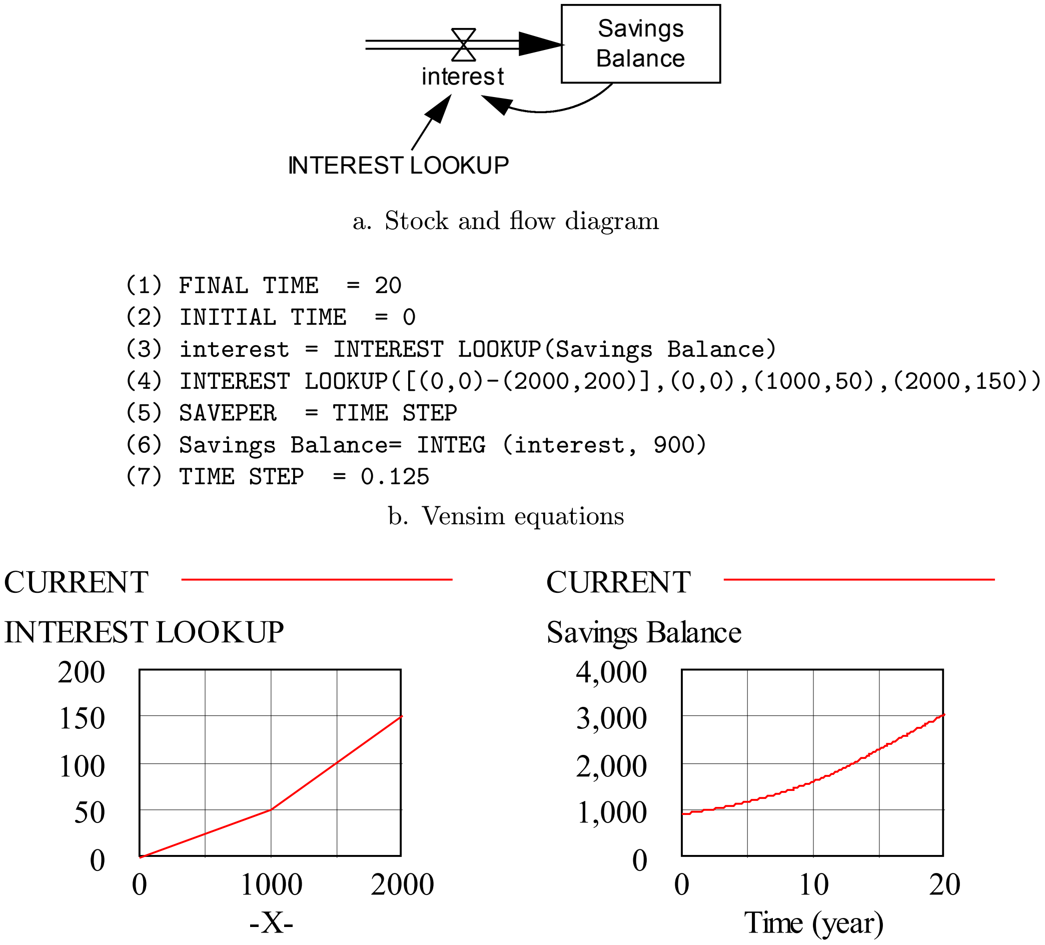
In addition to the IF THEN ELSE function, another approach to modeling nonlinear responses is provided by many simulation languages using lookup functions. A Vensim model for the money market account example which uses a lookup function is shown in Figure 8.3. With this approach, the nonlinear response function (which is interest for this example) is modeled by entering several pairs of points. The simulation program then creates a curve through these points which is used to determine the necessary values to run the simulation.

Equation (4) of Figure 8.3b defines this lookup function, which is called INTEREST LOOKUP. This function is specified by the three pairs of points (0, 0), (1000, 50), and (2000, 150). These points specify that there is $0 of interest per year earned on a Savings Balance of $0, $50 of interest earned per year on a Savings Balance of $1,000, and $150 of interest earned per year on a Savings Balance of $2,000. In Vensim, the lookup function calculates intermediate values by drawing straight lines between the specified pairs of values. Thus, the complete lookup function is shown in Figure 8.3c.

A casual examination of the Figure 8.2 and Figure 8.3 models indicates that these are the same, and thus they should show the same Savings Balance. However, a comparison of Figure 8.2c with Figure 8.3d shows that the Savings Balance curves are somewhat different. What has happened?

The difference between the curves shown in Figure 8.2 and Figure 8.3 illustrates a potential difficulty with using lookup functions. The lookup function in equation (4) of Figure 8.3b is specified over a range of values for Savings Balance from $0 to $2,000. However, the actual Savings Balance exceeds $2,000 during the thirteenth year. The specified behavior for a lookup function in Vensim when the range is exceeded is to clamp the output of the function at the highest specified value. Thus, whenever the Savings Balance is above $2,000, the lookup function INTEREST LOOKUP gives an output of $150. Clearly, this is incorrect for this savings account! (Vensim generates a warning message whenever the range specified for a lookup function is exceeded. In this particular case the following message is generated: `WARNING: At 13,25 Above INTEREST LOOKUP computing interest.`)

To correct this problem, it is necessary to widen the range over which INTEREST LOOKUP is specified. This can be done by replacing equation (4) in Figure 8.3b with the following:



a. Stock and flow diagram

```
(1) FINAL TIME  = 20
(2) INITIAL TIME  = 0
(3) interest = INTEREST LOOKUP(Savings Balance)
(4) INTEREST LOOKUP([(0,0)-(2000,200)],(0,0),(1000,50),(2000,150))
(5) SAVEPER  = TIME STEP
(6) Savings Balance= INTEG (interest, 900)
(7) TIME STEP  = 0.125
```

b. Vensim equations

c. Lookup function

d. Savings balance

Figure 8.3 *Model for a money market savings account (lookup function)*

```
(4) INTEREST LOOKUP([(0,0)-(5000,500)],(0,0), (1000,50),(5000,450))
```

which expands the upper limit for the range of INTEREST LOOKUP from a Savings Balance of $3,000 up to $5,000. When this change is made, identical output is generated to that shown in Figure 8.2c.

Comparison of IF THEN ELSE and Lookup Functions

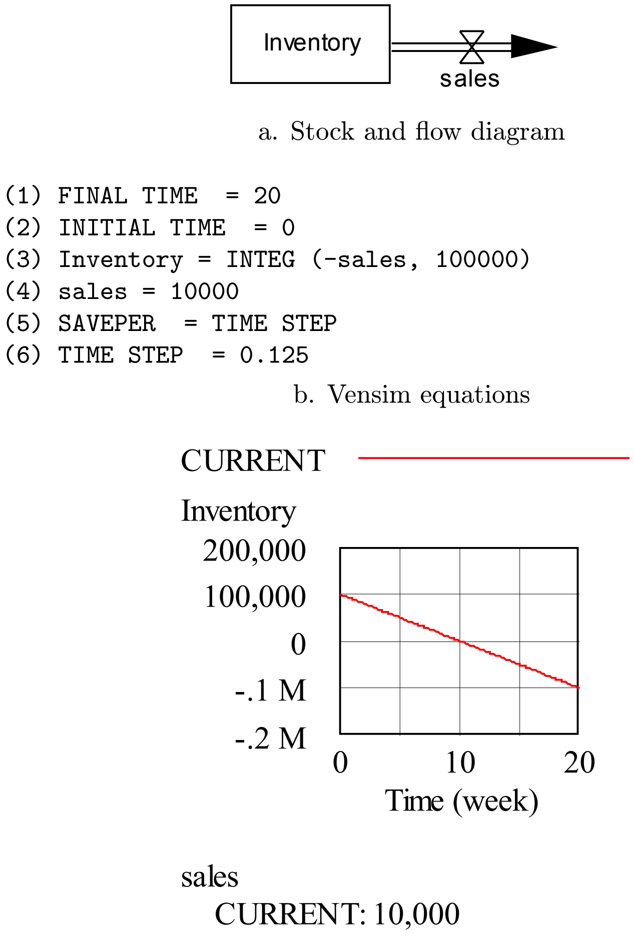
The IF THEN ELSE and lookup functions each have advantages and disadvantages for modeling nonlinear functions. As Figure 8.2b shows, it is possible to include constants in an IF THEN ELSE function so that a sensitivity analysis can be conducted directly in terms of the model constants BREAKPOINT, LOW RATE, and HIGH RATE using the automated procedures in Vensim. This cannot be done so directly when a lookup function is used. (Vensim does support a sensitivity analysis feature where a lookup function can be temporarily changed for a particular model run, but some calculation is necessary to determine exactly how the lookup function points must be changed to represent particular low and high interest rates for the money market account model.)

On the other hand, a lookup function can easily be constructed for situations where there are more than one breakpoint. While this can be done with IF THEN ELSE functions by nesting them, this can lead to complex function expressions.

8.2 Resource Constraints

Another common cause of nonlinear responses in a business process is resource constraints, such as limits on available personnel or production capacity. Figure 8.4 illustrates a first attempt at a model for a simple situation of this type where there is a fixed Inventory of 100,000 units available to sell, and a sales rate of 10,000 units per week. As the curves in Figure 8.4c demonstrate, this first model is inadequate. The Inventory drops to zero at ten weeks, but sales continue undiminished at a rate of 10,000 units per week. While an order backlog might grow for a while when the product is not available, sales are likely to drop as customers cannot get the product. The constraint on the number of units available to sell needs to be included in the model.

Figure 8.5 shows a modification to the Figure 8.4 model that uses an IF THEN ELSE function to shut off sales when Inventory reaches zero.



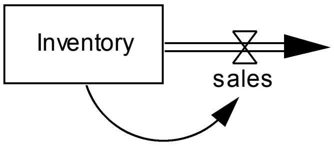
a. Stock and flow diagram

```
(1) FINAL TIME  = 20
(2) INITIAL TIME  = 0
(3) Inventory = INTEG (-sales, 100000)
(4) sales = 10000
(5) SAVEPER  = TIME STEP
(6) TIME STEP  = 0.125
```

b. Vensim equations

c. Inventory and sales

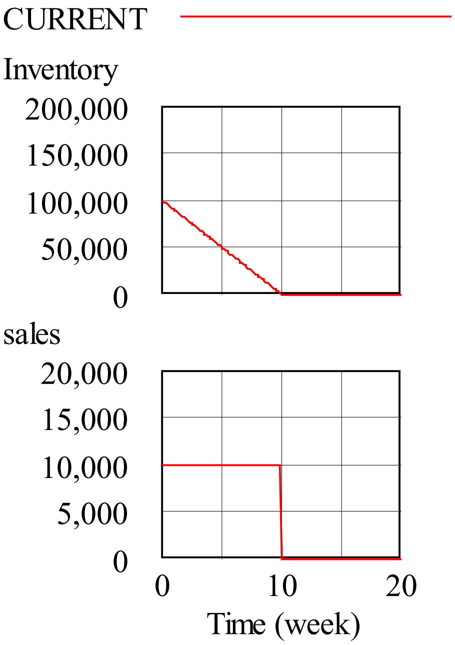
Figure 8.4 *Initial model for sales*



a. Stock and flow diagram

```
(1) FINAL TIME  = 20
(2) INITIAL TIME  = 0
(3) Inventory = INTEG (-sales, 100000)
(4) sales = IF THEN ELSE(Inventory > 0, 10000, 0)
(5) SAVEPER  = TIME STEP
(6) TIME STEP  = 0.125
```

b. Vensim equations



c. Inventory and sales

Figure 8.5 *Modified model for sales*

CHAPTER 9

Initial Conditions

Each stock in a model must be given an initial value. Once this is done, the simulation program will calculate the time history of each model variable for the specified time period. Sometimes initial values for the stocks can be quickly determined, but in other situations this task can be complicated and tedious. This chapter discusses two specific difficulties that often arise in specifying initial conditions for the stocks in a model.

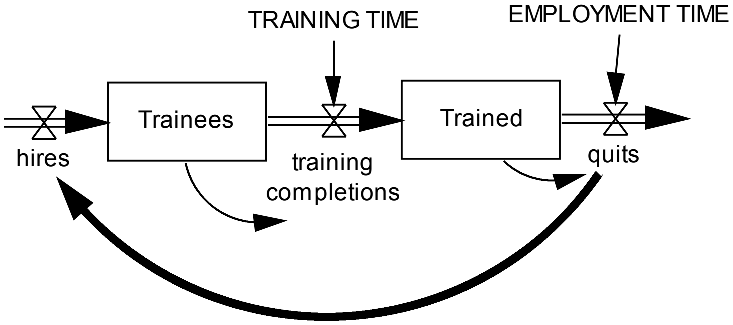
9.1 Initializing a Model to Equilibrium

Many models are specified for a process that is in *equilibrium.* That is, the values of the variables in the process are not changing. Often a model is being constructed for the process in order to estimate the impacts of making changes to the structure or operating policies of the process, and in such situations the first step in the modeling effort is to set the model up so that it reproduces the behavior of the existing process. This requires that the model be in equilibrium.

Figure 9.1 illustrates a simple model of this type for a personnel process which includes trainees and trained personnel. Trainees require an average of 3 months to train, and they stay in employment for an average of 3 years (36 months) once they are trained. Hiring is done to replace trained employees who quit, and the number of people to hire is determined using an exponential average of the quits over the last 6 months (26 weeks). It is known that there are currently 1000 trained employees, and it is estimated that there are 250 trainees.

When the model is run, the curves shown in Figure 9.1c result. Since it is desired to have the model in equilibrium, clearly something is wrong since the number of trainees and trained personnel both change over time. If the model were in equilibrium, these would remain constant.

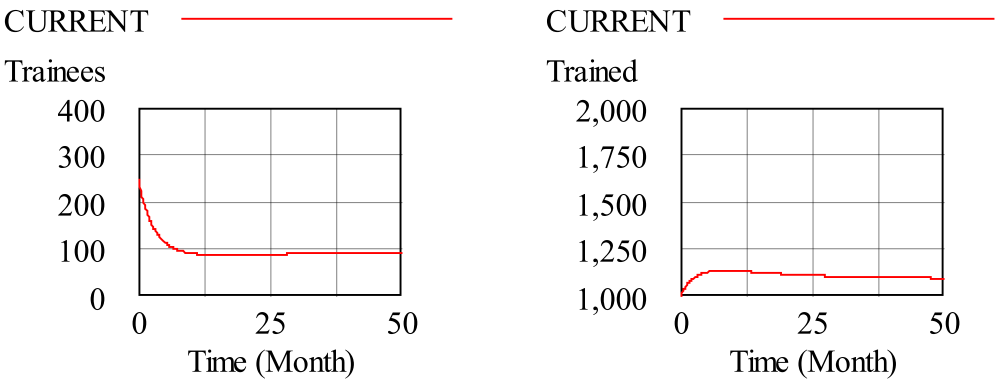
The difficulty has resulted from the process used to set the initial conditions for the two stocks Trainees and Trained. For the process to be in equilibrium, none of the model variables can change, and some thought shows that this means that *the inflows and outflows for each stock must be equal.* For the model in Figure 9.1, this means that hires must be equal to training completions (for



a. Stock and flow diagram

```
(01) EMPLOYMENT PERIOD = 36
(02) FINAL TIME  = 50
(03) hires = SMOOTH(quits, 26)
(04) INITIAL TIME  = 0
(05) quits = Trained / EMPLOYMENT PERIOD
(06) SAVEPER  = TIME STEP
(07) TIME STEP  = 0.125
(08) Trained = INTEG (+training completions-quits, 1000)
(09) Trainees = INTEG (hires-training completions, 250)
(10) training completions = Trainees / TRAINING PERIOD
(11) TRAINING PERIOD = 3
```

b. Vensim equations



a. Trainees

b. Trained

Figure 9.1 *Personnel training model*

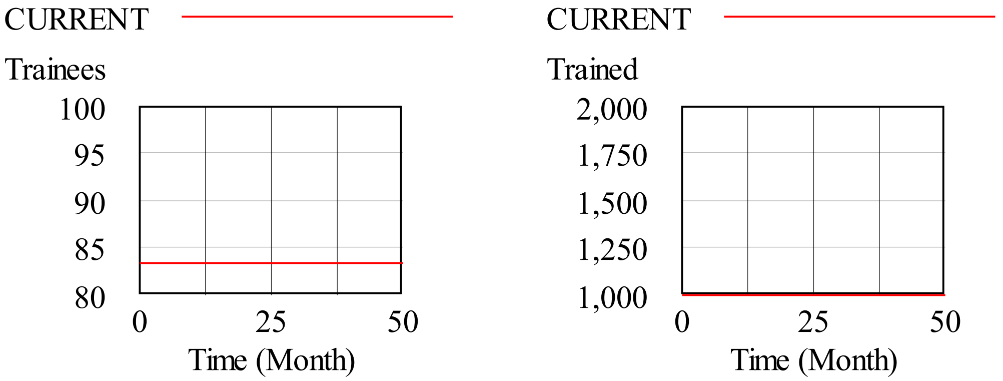


Figure 9.2 *Personnel training model (modified initial conditions)*

the inflows and outflows to Trainees to be equal), and training completions must be equal to quits (for the inflows and outflows to Trained to be equal.

However, as equations (03) and (10) in Figure 9.1b show, the flow training completions is equal to Trainees/TRAINING TIME, and the flow quits is equal to Trained/EMPLOYMENT TIME. Therefore, it is not possible to set both of the stocks Trainees and Trained independently. Specifically, for the rates training completion and quits to be equal it must be true that

$$\frac{\text{Trainees}}{\text{TRAINING TIME}} = \frac{\text{Trained}}{\text{EMPLOYMENT TIME}}$$

Thus, for the values of Trained, TRAINING TIME, and EMPLOYMENT TIME specified above, it must be true that Trainees = 1 000 × (3 36) = 83 3, which is considerable different from the value of 250 that was assumed in the Figure 9.1 model.

Rather than doing all this arithmetic by hand, it may make sense to enter the equation for the initial value of Trainees into the equation for this stock. This requires replacing equation (09) in Figure 9.1b with

```
(09) Trainees = INTEG (hires-training completions,
                    Trained * (TRAINING PERIOD / EMPLOYMENT PERIOD))
```

The results of making this change to the Figure 9.1 model are shown in Figure 9.2. Now the model is in equilibrium.

We have not discussed how to set the initial value for the flow hires. This is an exponentially smoothed value of quits. The SMOOTH function in Vensim is specified to have an initial output value equal to its input value. The initial value of the input to this SMOOTH is the initial value of quits, and so the initial value of hires (the output of the SMOOTH function) will be equal to the initial value of quits, which is (by the analysis above) equal to the initial value of training completions. Thus, the initial values of hires and training completions are equal, and hence the inflow and outflow for Trainees are equal, as they must be if the process is to be in equilibrium.

9.2 Simultaneous Initial Conditions

A second type of difficulty that can arise in specifying initial conditions involves the presence of simultaneous equations. Vensim and other similar simulation programs cannot solve simultaneous equations, and therefore it is not possible to set up initial conditions which require solving simultaneous equations. Such simultaneous equations can occur when a causal loop structure in a model includes only auxiliary variables. Figure 9.3 illustrates a model with this difficulty.

This is a model where the quality of a product is determined by the testing that is done on the product, and the order rate is impacted by the quality perceived by the customer. When the quality perceived by the customer is equal to one, there is an order rate of 10,000 units per month. In order to provide a product with a quality of one, each unit shipped must receive a testing effort equal to one hour of testing. Thus, at a shipping rate of 10,000 units per month, a TESTING CAPACITY of 10,000 hours per month is required. The quality perceived by the customer is an exponential smooth of the actual quality of the product shipped with a smoothing period of 6 months. That is, the customer perception of product quality changes more slowly than the rate at which actual product quality changes.

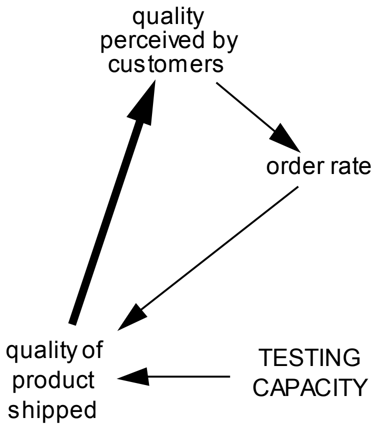
Everything appears to be correct in this model for it to be initialized to equilibrium. The TESTING CAPACITY is equal to 10,000 hours per month, which is the capacity required to maintain a product quality equal to one, which in turn is the quality required to have an order rate of 10,000 units per month. Hence it appears that the model in equilibrium. However, when you attempt to run the model, Vensim provides the following message: Model has errors and cannot be simulated. Do you want to correct the errors? If you click the Yes button, you see a further message that says there are Simultaneous initial value equations.

The difficulty is that there are simultaneous initial conditions involving the variables quality perceived by customers, order rate, and quality of product shipped. When Vensim attempts to solve for any of these variables, it ends up circling around back to the same variable. We know that the conditions specified in the Figure 9.3b equations are consistent, and they imply an order rate of 10,000 units per month with a quality of one, but Vensim is not able to determine this.

The SMOOTHI function is provided by Vensim to handle such situations. This has the same functionality as the SMOOTH function except that it allows you to specify an initial value for the output of the function. (Recall that with the SMOOTH function the initial output value of the function is equal to the initial input value.) The difficulty with simultaneous initial values is resolved by replacing equation (05) in Figure 9.3b with the following:

```
(5) quality perceived by customers =
        SMOOTHI(quality of product shipped, 6, 1)
```

This specifies that the initial output for the SMOOTH function will be equal to one. Vensim can then use this initial output to set the value for quality perceived by customers to one, and then can use this to determine the initial values for the other variables.



a. Stock and flow diagram

```
(1) FINAL TIME  = 50
(2) INITIAL TIME  = 0
(3) order rate = 10000 * quality perceived by customers
(4) quality of product shipped = TESTING CAPACITY /order rate
(5) quality perceived by customers
      = SMOOTH(quality of product shipped, 6)
(6) SAVEPER  = TIME STEP
(7) TESTING CAPACITY = 10000
(8) TIME STEP  = 0.125
```

b. Vensim equations

Figure 9.3 *Quality impacts ordering*