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.

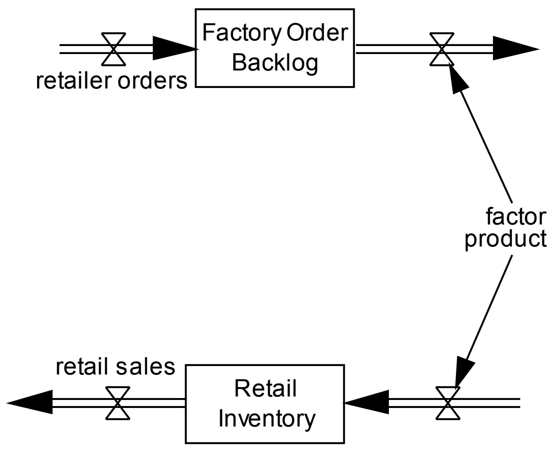


**Figure 5.1** *A simple production-distribution system*

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

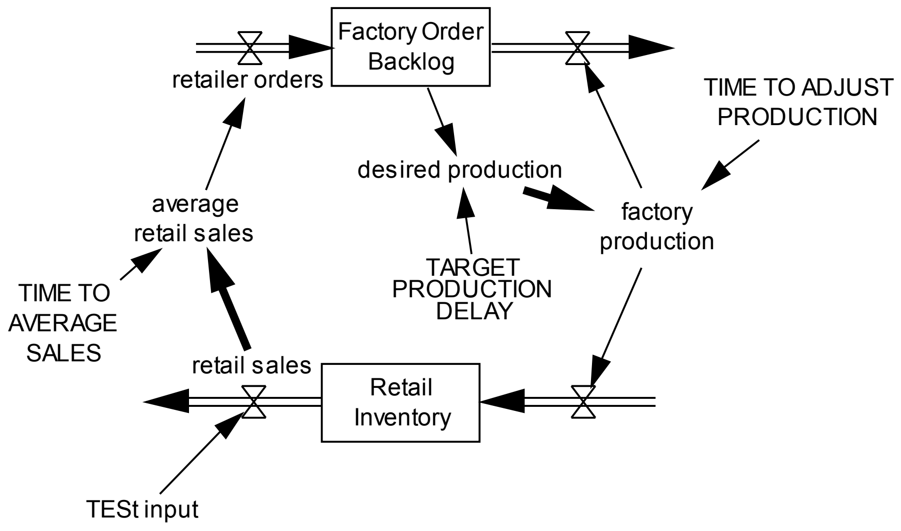


**Figure 5.2a** *Stock and flow diagram for first model*

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

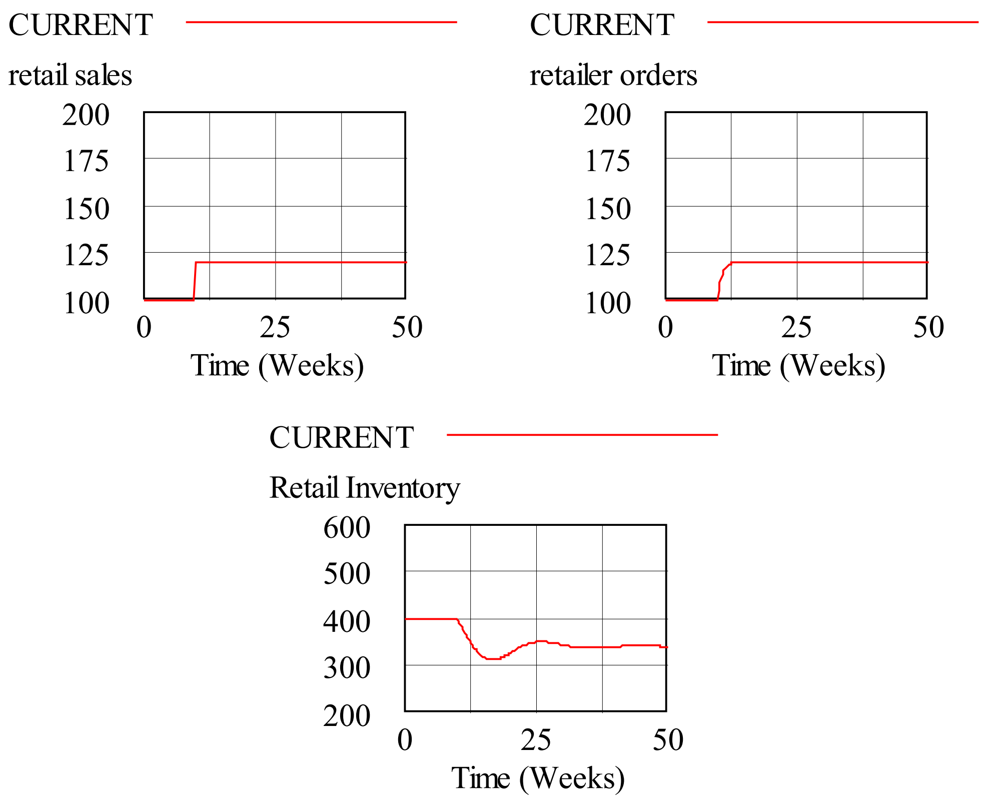


**Figure 5.2c** *Plots for first model*

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.

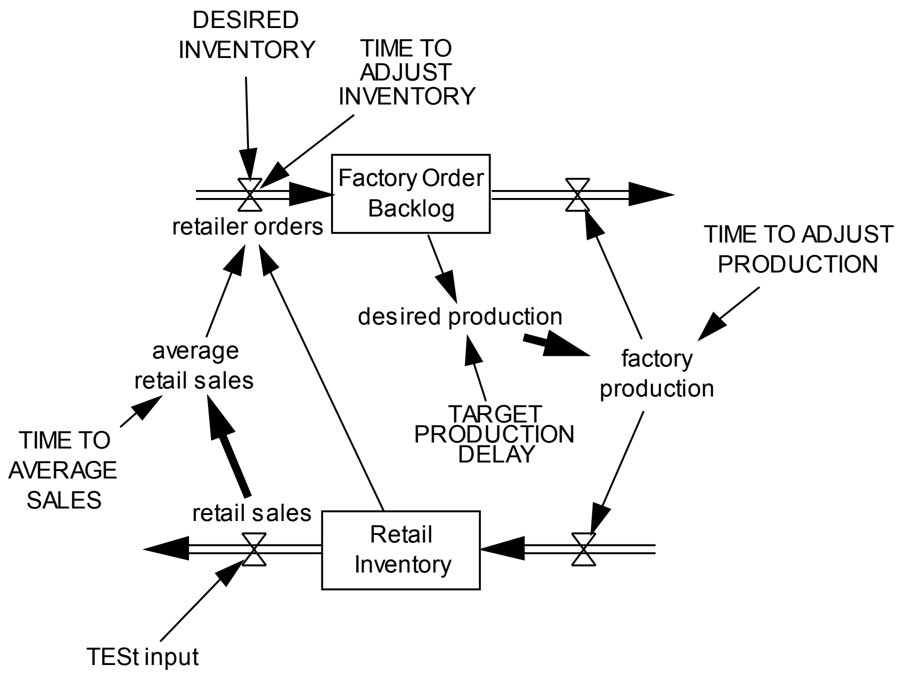


**Figure 5.3a** *Stock and flow diagram for second model*

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

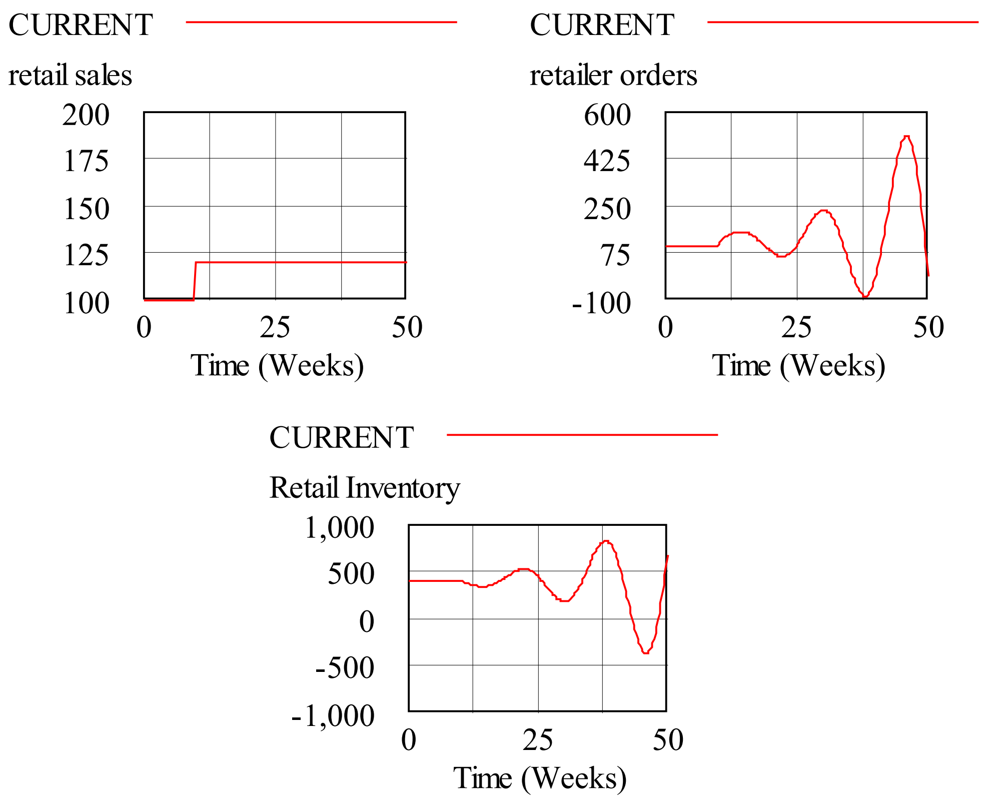


**Figure 5.3c** *Plots for second model*

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 Time + 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 Time + 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

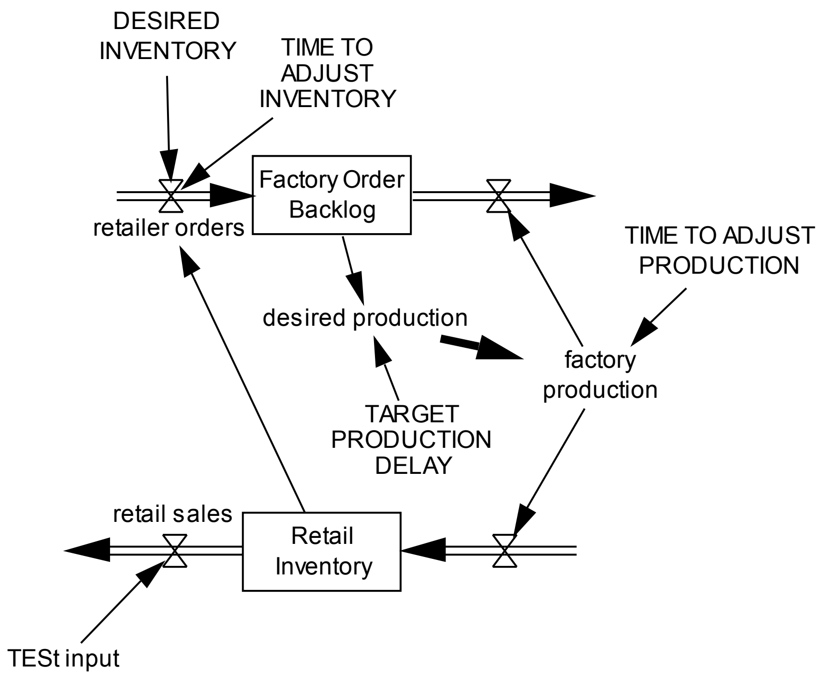


**Figure 5.4a** *Stock and flow diagram for third model*

```
(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*

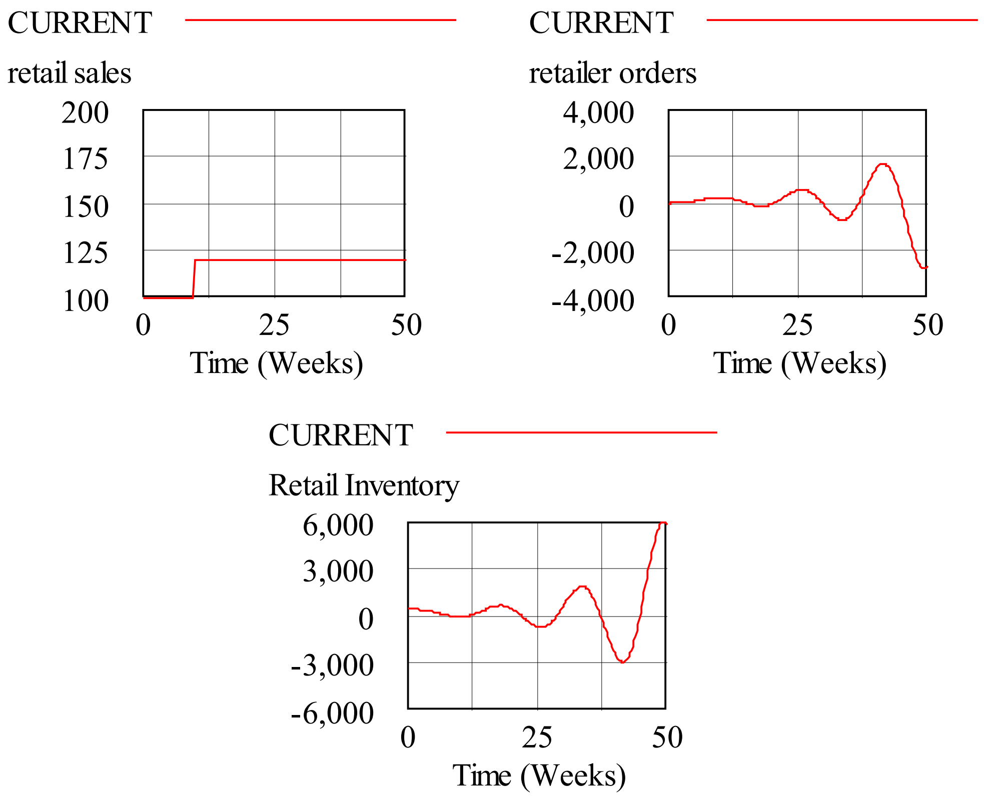


**Figure 5.4c** *Plots for third model*

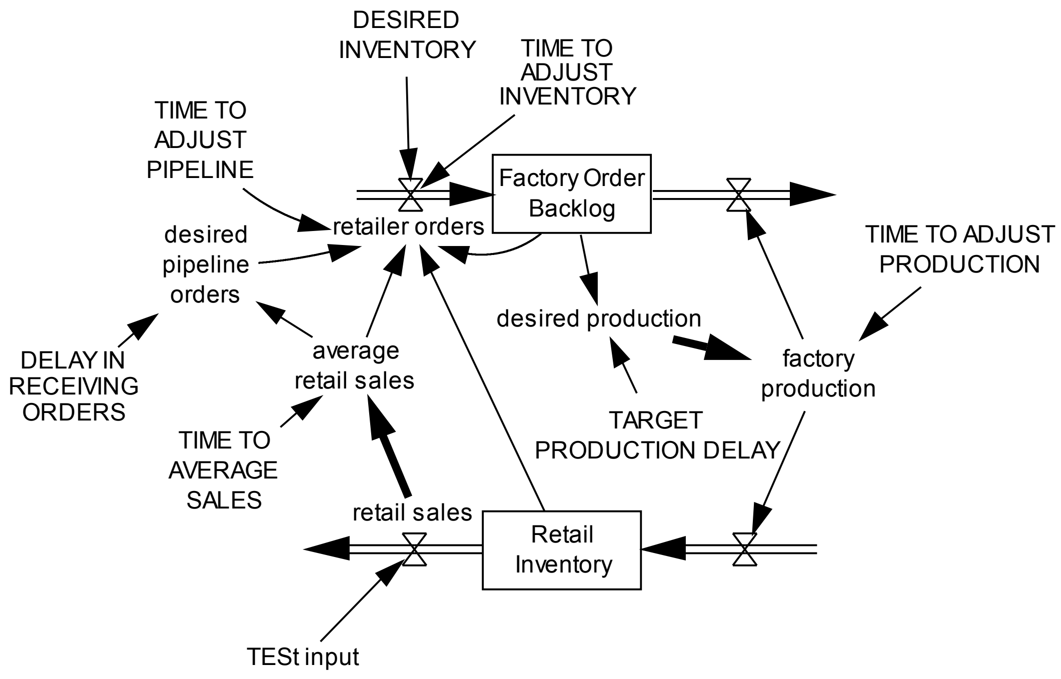


**Figure 5.5a** *Stock and flow diagram for fourth model*

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*

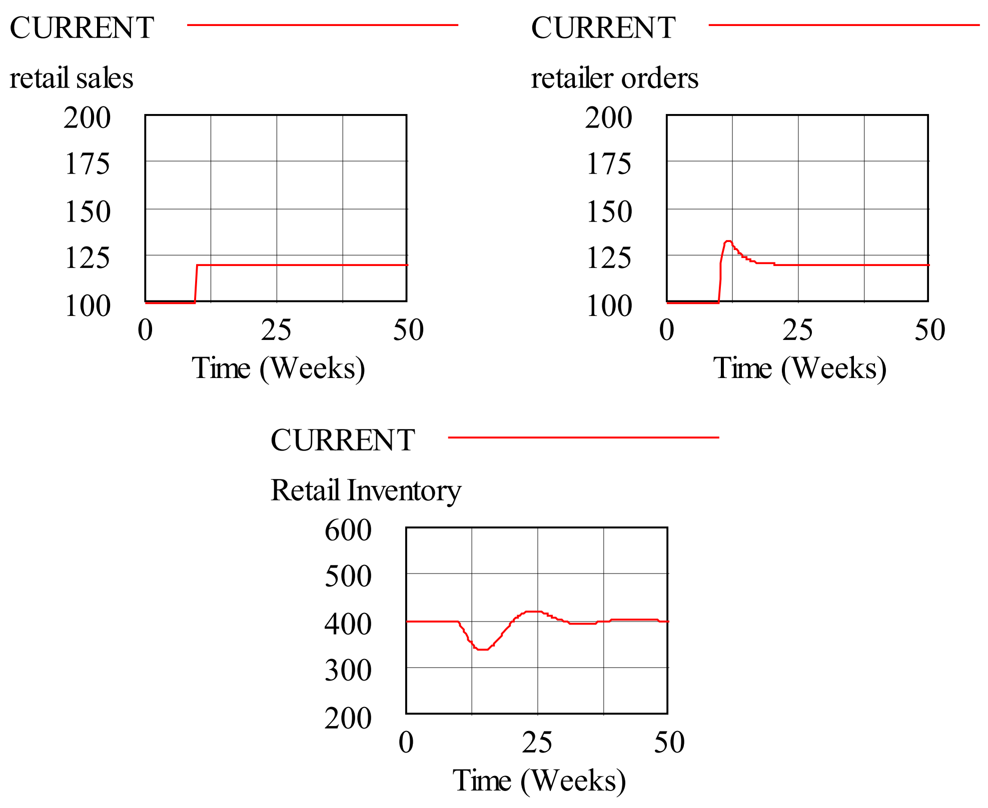


**Figure 5.5c** *Plots for fourth model*

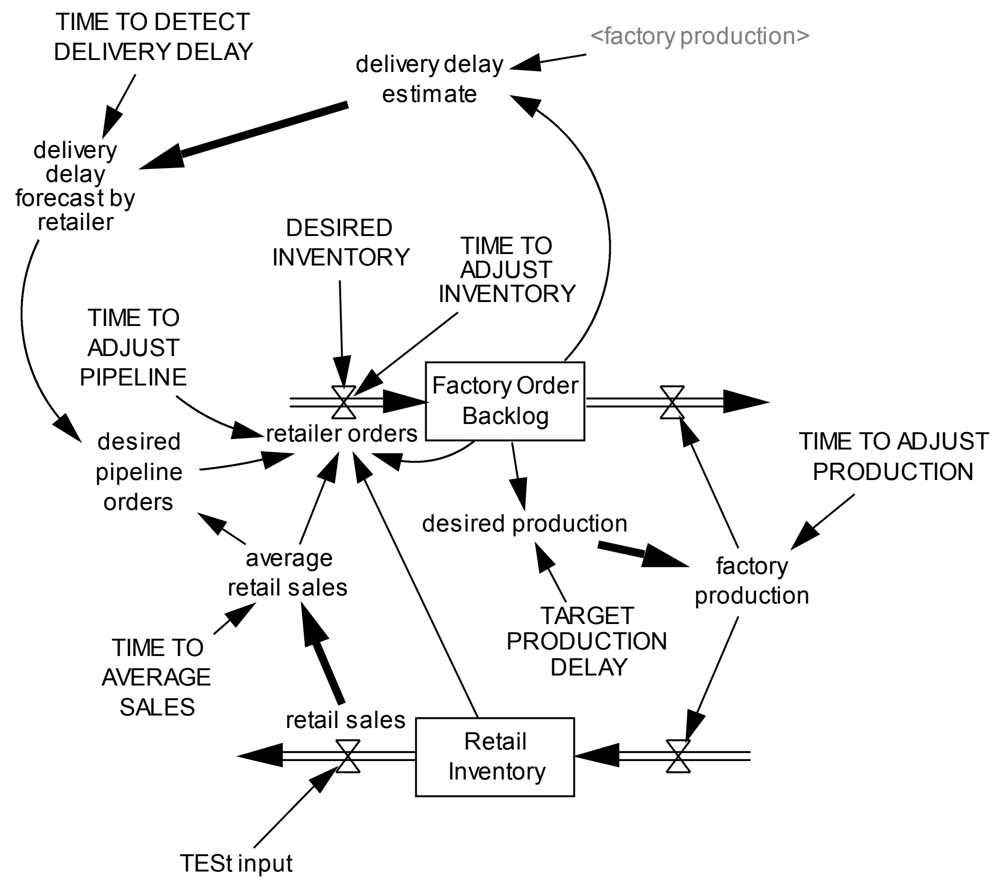


**Figure 5.6a** *Stock and flow diagram for fifth model*

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

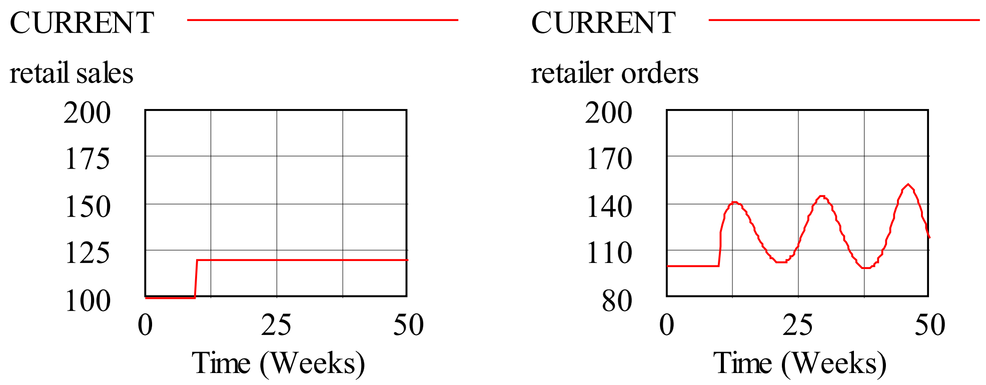
$$\text{TESt input} = 100 + \text{STEP}(20\ \ 10) * \text{RANDOM UNIFORM}(0\ \ 1\ \ 0)$$

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*



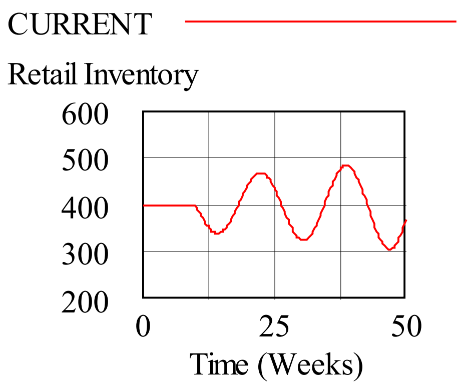


**Figure 5.6c** *Plots for fifth model*

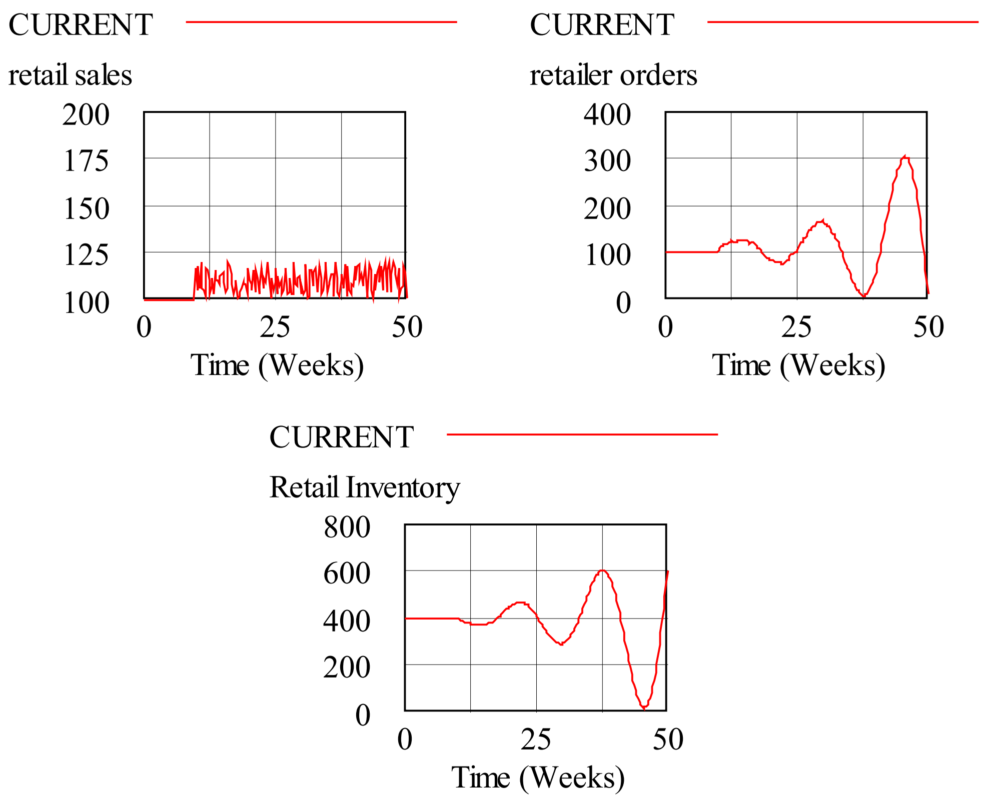
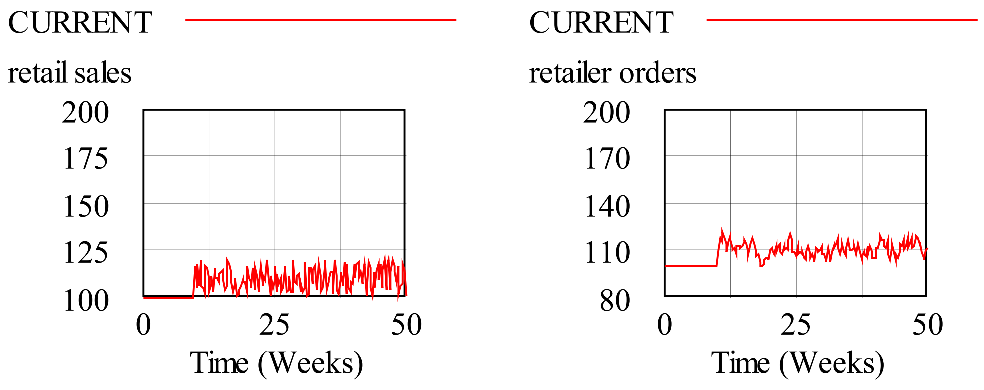


**Figure 5.7a** *Random retail sales for second model*



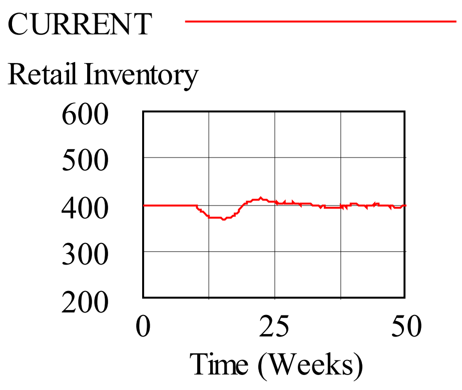


**Figure 5.7b** *Random retail sales for fourth model*

## 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.