The first day of the trend change, marked “1” in Figure 8-7, is very small, therefore no buy signal occurs. The changes on days 2 and 3, taken separately, are also smaller than the filter. Instead, on day 2 the total change from the 2nd day of the trend change to the recent low is compared with the filter, but that is still too small. A signal occurs on the third day, when the total difference from the low is greater than the filter, \(\text{AMA}(\text{low} + 3 \text{ days}) > \text{Filter}\).

The new rules for trading signals are

- **Buy** when \(\text{AMA} - @\text{lowest}(\text{AMA}, n) > \text{filter}\)
- **Sell** when \(@\text{highest}(\text{AMA}, n) - \text{AMA} > \text{filter}\)

When programming a computer, the sell signal may also be written

- **Sell** when \(\text{AMA} - @\text{highest}(\text{AMA}, n) < -\text{filter}\)

**Alternate Buy and Sell Rules.** It is difficult to record the recent high and low trend points on some computers and programmable trading machines. A simple, practical substitute is to compare the last three accumulated trend changes against the filter to generate a buy or sell signal.
This approach should work as well in all cases. For example,

Buy when $AMA - AMA[1] > \text{filter}$

or

Buy when $AMA - AMA[2] > \text{filter}$

or

Buy when $AMA - AMA[3] > \text{filter}$

**Testing the AMA**

Before using the Adaptive Moving Average, it will be necessary to test each market. The following points should help:

1. The primary parameter is the number of days used to calculate the Efficiency Ratio. This will be near 10 for the fast trader. Using a value below 5 will cause the ratio to jump from 0 to 1 quickly. Using a much larger value will cause the ratio to be more stable noise relationship that can be very attractive to the position trader.

2. The filter value is expressed as a percentage of the standard deviation of the trend changes; therefore, it is independent of price. However, a larger or smaller filter percentage is used to change the length of a trade. A small value allows an entry sooner, while a larger percentage will delay entry.

3. The number of days in the standard deviation, which determines the filter, could be fixed at 20. A statistical measure requires at least 20 days to have some stability.

Testing for short-term trading could fix the AMA days at 10 and the standard deviation days at 20, and test only the filter. Fewer parameters mean a more dependable solution. Longer-term positions are not affected by the filter; therefore it can be fixed at some value under 1.0.

**Profit-Taking.** Another look at Figure 8-6 shows that the Efficiency Ratio peaks over .80 (panel b) and the moving average days drops under 10 (in panel c) at points that would be good for taking profits. It is a characteristic of the Adaptive Moving Average that a high value for the Efficiency Ratio cannot be sustained and will be followed by a reversal. It would be best simply to take profits whenever the value exceeds a preset level. That threshold will vary based on the intrinsic noise of the market.
Programming the Adaptive Moving Average

The Adaptive Moving Average can be programmed into any spreadsheet or strategy-testing software. The following examples show the codes for Quattro Pro (very similar to Lotus), Telerate’s TeleTrac, and Omega’s TradeStation. Signals should not begin for 25 days, because the filter requires 20 days of AMA trendline changes, and the AMA needs an additional 5 days to start up.

Spreadsheet Instructions

Box 8-3 and Table 8-1 give the spreadsheet instructions and sample results. All constants have been placed in row 2. The recent AMA highs and lows are recorded in columns L and M.

Telerate’s TeleTrac

The TeleTrac code (Box 8-4) uses the alternate buy and sell signal calculation, comparing 3 days of AMA changes separately. It also uses the MACD study (“SIGNAL”) to calculate an exponential moving average with a changing smoothing constant. This code can be used to trade live data. Realized profits and losses are shown in the last line.

Omega’s Easy Language

The TradeStation code is the same for both System Writer and other Omega products (Box 8-5). Figure 8-8 shows the TradeStation display using the “AMA” system (Part 1) to give signals, the “AMA” indicator (Part 2), scaled to price, to plot the AMA trendline with the price chart, and “AMA smooth” (Part 3) to place the smoothing constant along the bottom of the graph.
Box 8-3. CALCULATING THE ADAPTIVE MOVING AVERAGE USING A SPREADSHEET

The Adaptive Moving Average can be easily calculated using a spreadsheet. In Table 8-1, columns B and C are input and the rest are calculated. Values begin in row 5. Repeated calculations are shown for row 15 (row 5 is for initialization). The AMA calculations (E-H) begin in row 15, because they require 10 days of data for the first value.

<table>
<thead>
<tr>
<th>Cell</th>
<th>Title</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>2-day smoothing constant</td>
<td>=2/(2+1)</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>30-day smoothing constant</td>
<td>=2/(30+1)</td>
<td></td>
</tr>
<tr>
<td>J2</td>
<td>Filter percentage factor</td>
<td>.10</td>
<td></td>
</tr>
</tbody>
</table>

**Constants:**

<table>
<thead>
<tr>
<th>B5</th>
<th>Date</th>
<th>Daily date</th>
<th>[Input data]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5</td>
<td>DMZ92</td>
<td>Deutsche mark price</td>
<td>[Input data]</td>
</tr>
<tr>
<td>I5</td>
<td>AMA(t)</td>
<td>(Use price C5 until row I5)</td>
<td>[Input data]</td>
</tr>
<tr>
<td>L5</td>
<td>Lows</td>
<td>Recent AMA lows</td>
<td>+I5</td>
</tr>
<tr>
<td>M5</td>
<td>Highs</td>
<td>Recent AMA highs</td>
<td>+I5</td>
</tr>
</tbody>
</table>

**Initial Values:**

**Repeated Formulas:**

- **D15** | D(t) | 10-day momentum | +C15-C5 |
- **E6** | 1DP | Positive 1-day volatility | =ABS(C6-C5) |
- **F15** | V(t) | 10-day volatility | =SUM(E6..E15) |
- **G15** | ER(t) | Efficiency Ratio | =ABS(D15/F15) |
- **H15** | C(t) | Smoothing constant | (G15*(S$2-S$2)+S$2)^2 |
- **I15** | AMA(t) | Adaptive Moving Average | +I4+H15*(C15-I14) |
- **J6** | DAMA | Difference in AMA | +I5-I5 |
- **K15** | Filter | Entry/exit filter | =STD(J15..J6)*$JS2 |
- **L6** | Lows | Recent AMA low | =IF(I15<I6,I6,L5) |
- **M6** | High | Recent AMA high | =IF(I5>I6,I6,M5) |
- **N15** | Buy | New buy signal | =IF(I15-L16>K15,'BUY',") |
- **O15** | Sell | New sell signal | =IF(M15-I15<K15,'SELL',") |

*Indicates to copy the contents of the cell down.

@ Indicates a Quattro function.
<table>
<thead>
<tr>
<th>DATE</th>
<th>Abs</th>
<th>10-Day</th>
<th>Smooth</th>
<th>1st Diff</th>
<th>E/X</th>
<th>AMA(t)</th>
<th>Trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>003601</td>
<td>6063</td>
<td>6063</td>
<td>6063.0</td>
<td>6063.0</td>
<td>6063.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602002</td>
<td>6341</td>
<td>6041</td>
<td>-22</td>
<td>6041.0</td>
<td>6053.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602003</td>
<td>6065</td>
<td>6065</td>
<td>24</td>
<td>604.1</td>
<td>6056.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>802004</td>
<td>6078</td>
<td>6078</td>
<td>13</td>
<td>604.1</td>
<td>6078.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>902005</td>
<td>6114</td>
<td>6114</td>
<td>36</td>
<td>604.1</td>
<td>6114.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100206</td>
<td>6121</td>
<td>6121</td>
<td>7</td>
<td>604.1</td>
<td>6121.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110209</td>
<td>6106</td>
<td>6106</td>
<td>-15</td>
<td>610.6</td>
<td>6121.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120210</td>
<td>6101</td>
<td>6101</td>
<td>-5</td>
<td>610.1</td>
<td>6121.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130211</td>
<td>6106</td>
<td>6106</td>
<td>65</td>
<td>610.1</td>
<td>6166.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>140212</td>
<td>6169</td>
<td>6169</td>
<td>3</td>
<td>610.1</td>
<td>6169.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150215</td>
<td>6165</td>
<td>6165</td>
<td>132.0</td>
<td>26</td>
<td>216</td>
<td>6173.9</td>
<td>4.9</td>
</tr>
<tr>
<td>160216</td>
<td>6222</td>
<td>6222</td>
<td>181.0</td>
<td>27</td>
<td>221</td>
<td>6188.8</td>
<td>15.0</td>
</tr>
<tr>
<td>170217</td>
<td>6169</td>
<td>6169</td>
<td>121.0</td>
<td>36</td>
<td>233</td>
<td>6188.4</td>
<td>-0.4</td>
</tr>
<tr>
<td>180218</td>
<td>6124</td>
<td>6124</td>
<td>136.0</td>
<td>28</td>
<td>248</td>
<td>6192.4</td>
<td>4.0</td>
</tr>
<tr>
<td>190219</td>
<td>6185</td>
<td>6185</td>
<td>71.0</td>
<td>29</td>
<td>241</td>
<td>6192.0</td>
<td>-0.4</td>
</tr>
<tr>
<td>200220</td>
<td>6189</td>
<td>6189</td>
<td>330.0</td>
<td>45</td>
<td>353</td>
<td>6192.0</td>
<td>0.4</td>
</tr>
<tr>
<td>210223</td>
<td>6221</td>
<td>6221</td>
<td>115.0</td>
<td>12</td>
<td>255</td>
<td>6196.4</td>
<td>3.1</td>
</tr>
<tr>
<td>220224</td>
<td>6278</td>
<td>6278</td>
<td>177.0</td>
<td>57</td>
<td>307</td>
<td>6210.2</td>
<td>13.8</td>
</tr>
<tr>
<td>230225</td>
<td>6326</td>
<td>6326</td>
<td>160.0</td>
<td>48</td>
<td>290</td>
<td>6228.4</td>
<td>18.2</td>
</tr>
<tr>
<td>240226</td>
<td>6347</td>
<td>6347</td>
<td>178.0</td>
<td>21</td>
<td>308</td>
<td>6246.8</td>
<td>20.2</td>
</tr>
<tr>
<td>250229</td>
<td>6420</td>
<td>6420</td>
<td>225.0</td>
<td>73</td>
<td>355</td>
<td>6282.7</td>
<td>34.1</td>
</tr>
<tr>
<td>260230</td>
<td>6394</td>
<td>6394</td>
<td>172.0</td>
<td>26</td>
<td>354</td>
<td>6296.9</td>
<td>14.2</td>
</tr>
<tr>
<td>270270</td>
<td>6400</td>
<td>6400</td>
<td>214.0</td>
<td>6</td>
<td>324</td>
<td>6318.9</td>
<td>22.0</td>
</tr>
<tr>
<td>280272</td>
<td>6446</td>
<td>6446</td>
<td>232.0</td>
<td>46</td>
<td>342</td>
<td>6347.4</td>
<td>28.4</td>
</tr>
<tr>
<td>290276</td>
<td>6442</td>
<td>6442</td>
<td>257.0</td>
<td>4</td>
<td>317</td>
<td>6376.3</td>
<td>28.9</td>
</tr>
<tr>
<td>300277</td>
<td>6543</td>
<td>6543</td>
<td>334.0</td>
<td>101</td>
<td>394</td>
<td>6431.4</td>
<td>55.1</td>
</tr>
<tr>
<td>310278</td>
<td>6550</td>
<td>6550</td>
<td>329.0</td>
<td>7</td>
<td>368</td>
<td>6470.4</td>
<td>36.1</td>
</tr>
<tr>
<td>320279</td>
<td>6442</td>
<td>6442</td>
<td>194.0</td>
<td>108</td>
<td>440</td>
<td>6468.1</td>
<td>-2.4</td>
</tr>
<tr>
<td>330280</td>
<td>6516</td>
<td>6516</td>
<td>190.0</td>
<td>74</td>
<td>466</td>
<td>6472.7</td>
<td>4.6</td>
</tr>
<tr>
<td>340281</td>
<td>6597</td>
<td>6597</td>
<td>250.0</td>
<td>81</td>
<td>526</td>
<td>6480.0</td>
<td>15.3</td>
</tr>
<tr>
<td>350271</td>
<td>6568</td>
<td>6568</td>
<td>148.0</td>
<td>29</td>
<td>452</td>
<td>6469.9</td>
<td>5.0</td>
</tr>
<tr>
<td>360275</td>
<td>6580</td>
<td>6580</td>
<td>186.0</td>
<td>12</td>
<td>468</td>
<td>6501.0</td>
<td>8.0</td>
</tr>
<tr>
<td>370276</td>
<td>6610</td>
<td>6610</td>
<td>210.0</td>
<td>30</td>
<td>492</td>
<td>6512.3</td>
<td>11.3</td>
</tr>
<tr>
<td>380277</td>
<td>6682</td>
<td>6682</td>
<td>236.0</td>
<td>72</td>
<td>518</td>
<td>6531.7</td>
<td>19.5</td>
</tr>
<tr>
<td>390270</td>
<td>6537</td>
<td>6537</td>
<td>95.0</td>
<td>145</td>
<td>659</td>
<td>6531.9</td>
<td>0.1</td>
</tr>
<tr>
<td>400271</td>
<td>6552</td>
<td>6552</td>
<td>9.0</td>
<td>15</td>
<td>573</td>
<td>6532.0</td>
<td>0.1</td>
</tr>
<tr>
<td>410272</td>
<td>6563</td>
<td>6563</td>
<td>13.0</td>
<td>11</td>
<td>577</td>
<td>6532.2</td>
<td>0.2</td>
</tr>
<tr>
<td>420273</td>
<td>6573</td>
<td>6573</td>
<td>131.0</td>
<td>10</td>
<td>479</td>
<td>6534.3</td>
<td>2.1</td>
</tr>
<tr>
<td>430274</td>
<td>6498</td>
<td>6498</td>
<td>218.0</td>
<td>75</td>
<td>480</td>
<td>6534.0</td>
<td>-0.3</td>
</tr>
<tr>
<td>440276</td>
<td>6593</td>
<td>6593</td>
<td>24.0</td>
<td>95</td>
<td>494</td>
<td>6534.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>
### Box 8-4. TELETRAC CODE FOR THE ADAPTIVE MOVING AVERAGE

```plaintext
item  *DEUTSCHMRK_3/93
date  *DATA(date,first,last,item)
open  *DATA(open,item)
high  *DATA(high,item)
low   *DATA(low,item)
close *DATA(close,item)
vol   *DATA(vol,item)
oi    *DATA(oi,item)
period 10

diff  Abs_val(close-close[1])
noise Sum(diff,period)
signal close-close[period]
efratio Abs_val(signal)/noise
smooth Power(efratio^(.666-.0645)+.0645,2)
macd  close
ama   Signal(macd,smooth)
dama  ama-ama[1]
dama2 ama-ama[2]
dama3 ama-ama[3]
sdays 20
filter .10
sdama Std_dev(dama,sdays)*filter
buyif dama>sdama2>sdama3>sdama
buy   dama>=0&buyif
sellf dama<-sdama2<-sdama3<s-dama
sell  dama<0&sellif
strategy Trade(buy,sell,sell,buy)
realized Clos_PL(strategy,close,.0001,.0002)
```

The TeleTrac code uses a special function, Signal, to calculate a trendline based on a changing exponential smoothing constant. This code also compares three days of accumulated trend changes to avoid a gradual change that might not give a new signal. Period, filter, and sdays are entered as "coefficients" to allow for optimization.
Box 8-5. OMEGA EASY LANGUAGE CODE FOR THE ADAPTIVE MOVING AVERAGE*

Part 1: Enter as a "system."

```plaintext
inputs: period(10), filter(.1);
vars: noise(0), signal(0), diff(0), efratio(0), extlow(0), exthigh(0),
      smooth(1), fastend(.666), slowend(.0645), AMA(0);

{ CALCULATE EFFICIENCY RATIO }
diff = @AbsValue(close - close[1]);
if(currentbar < = period) then AMA = close;
if(currentbar > period) then begin
    signal = @AbsValue(close - close[period]);
    noise = @Summation(diff,period);
    efratio = signal/noise;
    smooth = @Power(efratio*(fastend - slowend) + slowend,2);

{ ADAPTIVE MOVING AVERAGE }
    AMA = AMA[1] + smooth*(close - AMA[1]);

{ TREND CHANGE FILTER FROM LAST TURN }
    if(AMA > AMA[1] and AMA[1] < AMA[2]) then extlow = AMA[1];
    if(AMA < AMA[1] and AMA[1] > AMA[2]) then exthigh = AMA[1];

{ TRADING SIGNALS }
    if(currentbar > period + 5) then begin
        if(AMA > AMA[1] and AMA - extlow > filter) then buy on close;
        if(AMA < AMA[1] and exthigh - AMA > filter) then sell on close;
        end;
    end;
```

Note that this code saves the most recent trend turning points as extlow and exthigh. It can then use those points to compare the accumulated change of direction against the filter and avoid missing a signal due to a very slow trend change.
Box 8-5. (Continued)

Part 2: Enter as an “indicator” to see the trendline on the chart page.
inputs: period(10);
vars: noise(0), signal(0), diff(0), efratio(0),
     smooth(1), fastend(.666), slowend(.0645), AMA(0);

{ CALCULATE EFFICIENCY RATIO }
diff = @AbsValue(close - close[1]);
if(currentbar <= period) then AMA = close;
if(currentbar > period) then begin
    signal = @AbsValue(close - close[period]);
    noise = @Summation(diff,period);
    efratio = signal/noise;
    smooth = @Power(efratio*(fastend - slowend) + slowend,2);
end;

{ ADAPTIVE MOVING AVERAGE }
AMAx = AMA[1] + smooth*(close - AMA[1]);
Plot1(AMAx,"AMA");
end;

Part 3: Enter as an “indicator” to plot the smoothing constant on the chart page.
inputs: period(10);
vars: noise(0), signal(0), diff(0), efratio(0),
     smooth(1), fastend(.666), slowend(.0645);

{ CALCULATE EFFICIENCY RATIO }
diff = @AbsValue(close - close[1]);
if(currentbar <= period) then AMA = close;
if(currentbar > period) then begin
    signal = @AbsValue(close - close[period]);
    noise = @Summation(diff,period);
    efratio = signal/noise;
    smooth = @Power(efratio*(fastend - slowend) + slowend,2);
    Plot1(smooth,"AMA smooth");
end;

Copyright 1993, P. J. Kaufman. All rights reserved.
Figure 9-6. Adaptive Moving Average for the Eurodollar on TradeStation.
9

Computer Learning, Neural Networks, and New Technology

Using a computer develops special skills in the same sense as driving a car or learning a new language. The process is more intimidating than it is difficult. New computer applications are much more user-friendly than they were five years ago. Most programs allow you to use a “mouse,” provide pop-up help screens, and explain each option on the screen whenever you point the mouse at a special icon symbol. To our relief, it is also harder to destroy a program by hitting the wrong keys. This removes some of the concerns about learning. If you press the Enter instead of the Esc key, the computer may tell you ENTRY INVALID. TRY AGAIN, or just ignore you.

Teaching a computer the rules for a trading strategy is a learning process for both you and the computer. Unlike the human brain, the computer cannot infer a meaning, it must be told precisely. We often think that we are very clear about giving instructions, but the section on “fuzzy logic” will show how many of our expressions are vague. The following section is intended to emphasize how exact you must be in specifying rules to get the computer to give the right answer. If you have never programmed trading rules, it will be well worth your time to follow this process through the next few pages.
The Teaching Process:  
First the Trainer

Because some important new technologies “teach” the computer how to solve a problem, and strategy-testing programs allow you to define specific rules, this chapter will show how human this process can be. An early marvel of engineering skill, the Erector Set, will be used to illustrate the steps (Box 9-1).

The basic Erector Set has flat metal pieces, screws, and bolts. Our sample set uses only nine pieces, conveniently designed to fit together, as shown in Figure 9-1, with the pieces numbered for reference. It has 18 sets of bolts and nuts, exactly \( \frac{1}{2} \) the number of total corners.

Your task is to build a flat, narrow 4-inch bridge long enough to cover a 33-inch span. You must write the rules for constructing the bridge clearly enough for someone else to follow them exactly. You can start with the five rules given in the first panel of Box 9-1.

Applying Prices to the Training Game

When we apply the same process to finding a sequence of prices that results in a net move of +50 points in the DJIA, some interesting similarities and differences appear. If we think of each “piece” as a price, we can restate some of the rules:

**RULE A.** Begin with a partially constructed bridge of length 1 (the original investment).

**RULE B.** Decide on the length of the final bridge before starting.

**RULE C.** You must take the prices (“pieces”) in the order they come, but you may discard them if they are too small (but only if you have decided in advance, what is “too small”).

Discarding a small piece is a threshold criterion that will be used later in this chapter in the section, “Neural Networks.” It allows us to decide that a very small price is not relevant for determining a new trading signal.

**RULE D.** Prices that exceed the threshold are always attached at the same end (of the price series).

**RULE E.** A price piece can either add or remove (positive or negative) length:

**RULE F.** The length cannot become less than \( \frac{1}{2} \) the original size (maximum loss rule).
**RULE G.** The bridge must be constructed by a certain time using a fixed number of pieces (return on investment rule).

Some additional considerations are:

- Price changes come in more sizes. Some are exceptionally big, and others are very small and can be ignored. When a very large positive piece appears, the goal is successfully reached; when a very large negative piece appears, the game is lost. The frequency of catastrophic loss is based on the number of occurrences of large negative pieces, or price shocks. Chapter 7 shows that there is no way to avoid these events, but there are ways to survive them.
- If too many small pieces are used, the time spent is no longer cost-effective.
- You cannot reach the goal in the scheduled time if too many pieces take away from its length.

Most important, when using prices instead of playing games,

- There *may not* be a solution within the limits of the rules, time, and objectives.

Once again: *There may not be a solution.* Forcing an answer from price patterns and data is not a solution. It may be necessary to look at the problem from an entirely different perspective.

Computers do not think; they simply follow your instructions. Writing the rules requires practice. When an instruction is missing, the answer is wrong, even when the results appear to be good. The only way to know that the computer has calculated everything correctly is to check the results manually for a few different cases. The more complex the trading strategy, the longer it will take to verify.

It is easy to make mistakes when specifying rules and typing formulas. It is not likely that any system has been written that did not require careful computer debugging. The following sections on new technologies will describe a number of interesting approaches to defining rules and making decisions.

**Artificial Intelligence and Pattern Recognition**

The field of artificial intelligence (AI) includes many new technologies for prices forecasting, such as expert systems, neural networks, and
Box 9-1. THE ERECTOR SET PIECES

Figure 9-1. Erector set pieces.

Erector set pieces

There are nine pieces, forming four sets of plates each with an area of 4" × 8". Each plate is fully covered with holes. There are 18 bolts and 18 nuts.

Objectives

Can you construct a bridge that is 33" long and 4" wide at all points? Can you teach someone else by writing the exact rules for doing this the first time? The following five rules will get you started:

RULE 1. You must take the pieces in any order.
RULE 2. Two flat pieces can be attached by inserting a bolt through a hole in each of the two pieces, and fastening it with a nut.
RULE 3. The bridge must be 4" wide at all places.
RULE 4. All corners must be fastened except the four at the two ends of the bridge.
RULE 5. The bridge must reach across a 33" span.

Teaching Yourself

To follow the rules, you select each piece with your eyes closed. You must write an unambiguous rule for using any piece before you see it. This is important because, when you test stock or futures prices, you do not know what will come next.

You take the first piece and look at it. You realize immediately that the five rules do not include instructions for the first piece. The rules require at least two pieces. You cannot continue without adding;
RULE 6. Start with two pieces; after the first move, take one piece at a time.

Starting again, you take two pieces and fasten them together by applying Rule 2. You realize that you do not know which holes to fasten.

RULE 7. Fasten the shortest sides together by overlapping the edges (Figure 9-2) so that the holes nearest the edges align; if the shorter edges are the same size, then attach them so that they are aligned at the ends; if they are different lengths, attach them so that one end is aligned.

Rule 7 will prevent bolting a small piece in the middle of a big piece, and will prevent the bridge from going around in a circle.

Beginning again, we draw pieces 8 and 4 and attach them end to end, into an “L”-shaped form, as in Figure 9-3. Notice that the total length is \( \frac{1}{4} \) inch less than the sum of the two pieces, because they overlap. Piece 1 comes next, but there are two choices: It can be attached to the end of piece 8 or the end of piece 4. This suggests two more problems that need rules:

RULE 8. Continue to add pieces in the same direction.

RULE 9. Before adding a piece at the very end of the ramp, first check to see if the piece fits a missing slot to finish the 4” width.

This process of adding rules based on experience is called learning by feedback. By continuing, we will eventually complete a bridge that spans the 33 inches with 1 inch overlapping each end. This lengthy process is identical to “teaching” the computer.

**Figure 9-2. Overlapping pieces.** Bolting two pieces together takes away \( \frac{1}{4} \) from the total length.

**Figure 9-3. Attaching the first three pieces together.** Extra rules are needed.
fuzzy logic. There is still a lot to learn from the first AI methods, which centered around basic pattern recognition.

Think about price changes as direction only:

up, up, down, up, unchanged, up, up, down,...

What comes next? If you were taking a school test, the answer would be up. If you were investing in the market, how much would you risk on the next day being up? Nothing, because the pattern was not repeated enough times, and markets are not expected to perform with such regularity.

The existence of a previous pattern that is identical to the current one is not enough to risk an investment. For example, you test 10 years of data and find 74 cases of identical 5-day patterns, either up, up, down, up, up or down, down, up, down, down. If the results are random, then this pattern will be followed by 37 up days and 37 down days. If there are 42 up days and 32 down days, would you consider buying each time this pattern appears? There may be an edge, but a very small one. Or there may not have been enough cases in the 10 years for the random distribution to have appeared.

Simple pattern recognition is a difficult tool to use in trading. Its success is entirely an issue of statistics, and must be treated in that light. The following methods of artificial intelligence are much more likely to produce good results.

**Applications of Expert Systems**

Applications of *artificial intelligence* are intended to have computers operate the way humans think. It may not be clear that the process is really desirable, but science considers the unachievable as a challenge, and sometimes pursues it without understanding why. The development of the following concepts began long before everyday technology could support them.

*Expert systems* have the very sensible goal of duplicating expert advice and decisions. This approach has had remarkable success in medical diagnosis and could be equally applicable to financial issues.

New technology tends to create terminology to express the ideas, and expert systems are no exception. Having the proper words seems to be part of the process:

- *Teaching the computer* refers to the act of entering data and rules into the machine, such as:
FACT 1: Bob's parents are George and Martha.
FACT 2: Mary's parents are George and Martha.
RULE 1: If you drive faster, you will get there sooner.
RULE 2: You can't drive faster than the speed limit.
RULE 3: If there is more traffic, you must drive slower.

- *Inference* means creating new facts from existing ones. Using facts 1 and 2, we get

INERENCE: Bob and Mary are brother and sister.

- *Pruning* is the sorting through of all the information to find the most relevant. Because the brain and the machine are filled with data and rules, it is necessary to select the ones that apply to the current problem. You would not want to answer the question, "Why did the stock market drop?" with "Because of the earthquake in Armenia." An interesting item, but unrelated.

- An *expert system* is one that deals with a special area, such as medical diagnosis, oil spills, or stock market forecasting. By listing all the facts and rules in the specialized *domain*, the system is expected to substitute for the benefits of a team of experts.

To create an expert system to make stock market decisions, first write the relationships that are facts, in any order. For example, Table 9-1 gives a set of 10 rules.

**Forward Chaining**

The definitions and rules in Table 9-1 combine to form a *knowledge base*. For convenience, the shortened names referring to data are explained in the list of *variable names*. Using a process called *forward chaining*, start with an important piece of information and follow one rule to another until you find your answer. For example, The Wall Street Journal front page reads:

FED CUTS RATE HALF POINT TO 4¼

and you want to know how the stock market should react. Rule 1 states that, IF interest FALLS, THEN stocks RISE. Therefore, expect the stock market to rally.

If the *Journal* had said:

DOLLAR DROPS AGAINST THE YEN

Then Rule 2 gives IF dollar FALLS, THEN interest RISE. That is chained to Rule 1, which states, IF interest RISE, THEN stocks FALL.
Table 9-1. Expert System Rules for Stock Market Decisions

<table>
<thead>
<tr>
<th>Rule</th>
<th>Statement of Rule (a)</th>
<th>Opposite Rule (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF THEN interest</td>
<td>IF interest stocks</td>
</tr>
<tr>
<td></td>
<td>FALL RISE</td>
<td>RISE</td>
</tr>
<tr>
<td>2</td>
<td>IF THEN U.S. dollar</td>
<td>IF RISE</td>
</tr>
<tr>
<td></td>
<td>interest</td>
<td>RISE</td>
</tr>
<tr>
<td>3</td>
<td>IF THEN inflation</td>
<td>IF inflation</td>
</tr>
<tr>
<td></td>
<td>interest</td>
<td>interest</td>
</tr>
<tr>
<td>4</td>
<td>IF THEN GNP</td>
<td>IF GNP</td>
</tr>
<tr>
<td></td>
<td>interest</td>
<td>interest</td>
</tr>
<tr>
<td>5</td>
<td>IF THEN Ger Bund rate</td>
<td>IF RISE</td>
</tr>
<tr>
<td></td>
<td>interest</td>
<td>Ger Bund rate</td>
</tr>
<tr>
<td>6</td>
<td>IF THEN p/cap spend</td>
<td>IF RISE</td>
</tr>
<tr>
<td></td>
<td>inventories</td>
<td>p/cap spend inventories</td>
</tr>
<tr>
<td>7</td>
<td>IF THEN unemployment</td>
<td>IF RISE</td>
</tr>
<tr>
<td></td>
<td>p/cap spend</td>
<td>unemployment</td>
</tr>
<tr>
<td>8</td>
<td>IF THEN inventories</td>
<td>IF RISE</td>
</tr>
<tr>
<td></td>
<td>production</td>
<td>inventories</td>
</tr>
<tr>
<td>9</td>
<td>IF THEN production</td>
<td>IF RISE</td>
</tr>
<tr>
<td></td>
<td>GNP</td>
<td>production</td>
</tr>
<tr>
<td>10</td>
<td>IF THEN Fed money sup</td>
<td>IF DEC</td>
</tr>
<tr>
<td></td>
<td>interest</td>
<td>Fed money sup</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest</td>
<td>U.S. interest rates</td>
</tr>
<tr>
<td>Stocks</td>
<td>U.S. stock market</td>
</tr>
<tr>
<td>U.S. dollar</td>
<td>U.S. dollar exchange rate</td>
</tr>
<tr>
<td>Inflation</td>
<td>Rate of U.S. inflation</td>
</tr>
<tr>
<td>GNP</td>
<td>The Gross National Product</td>
</tr>
<tr>
<td>Ger Bund rate</td>
<td>German Bund interest rate</td>
</tr>
<tr>
<td>P/cap spend</td>
<td>Per capita spending</td>
</tr>
<tr>
<td>Inventories</td>
<td>U.S. manufacturers' inventories</td>
</tr>
<tr>
<td>Production</td>
<td>Total U.S. manufacturing production</td>
</tr>
<tr>
<td>Fed money sup</td>
<td>Federal Reserve money supply target</td>
</tr>
</tbody>
</table>

If the news is:

**UNEMPLOYMENT RISES**

Start at Rule 7b, IF unemployment RISES, THEN p/cap spend FALLS,
Then to Rule 6b, IF p/cap spend FALLS, THEN inventories RISE,
To Rule 8b, IF inventories RISE, THEN production FALLS,
To Rule 9b, IF production FALLS, THEN GNP FALLS,
To Rule 4a, IF GNP FALLS, THEN interest FALLS,
Finally to Rule 1 IF interest FALLS, THEN stocks RISE.
By chaining one rule to another, the computer should reach the same conclusion as an expert. The only problem is that this answer does not make sense. In this case, although all the rules are perfectly correct, the conclusion *unemployment rises, therefore the stock market rises* is wrong.

To be fair, it is not technically wrong. What is missing is the time delay. Unemployment will cause interest rates to be lowered, which will move the stock market higher. But not on the same day. First, the market will drop on the news. Each sequence, represented by a rule, must be assigned a reaction time or completion criterion. Rule 7b should really read:

\[
\text{IF unemployment RISES, THEN p/cap spending FALLS over the next 3 months.}
\]

By adding time to each rule, we come closer to an expert system.

**Drawing on the Knowledge Base**

Once the knowledge base has been established, many different questions can be asked: "What is the effect on the stock market when the Fed wants to increase the money supply? What is the effect of an increase in unemployment on the U.S. dollar?"

The reverse of many items in the knowledge base, but not all, may also be used: "IF interest rates DO NOT FALL and the GNP IS NOT POSITIVE, THEN the stock market WILL NOT RISE."

**Resolving Conflicts of Multiple Events**

As easy as it is to show sequences stemming from single events, it is not realistic. Two or more significant factors are usually present and often conflict with one another. Which is more important, if any? The following rules can be added to help determine which government statistic is most important:

1. **W1.** IF actual statistic minus expectations IS MOST EXTREME THEN most important.

2. **W2.** IF cumulative difference of last 3 statistics minus expectations IS MOST EXTREME THEN most important.

3. **W3.** IF actual statistic minus year ago IS MOST EXTREME THEN most important.

4. **W4.** IF statistic minus long-term mean IS MOST EXTREME THEN most important.
But it is still not complete. Two or more events may be extreme in different ways. They may be cumulative or offsetting in their effect on the stock market. Rules must be written to resolve these factors. This conflict resolution can be the weak point of expert systems.

**Validation**

When it is all in the computer and an answer pops out, how do you know it is correct? Because the process is logical, the individual steps can be traced to prove the answer. But the example of "UNEMPLOYMENT RISES" gave the right answer but not the right time. In the end, the final decision is yours. The answer must seem right, and satisfy the test of reasonableness.

**Neural Networks**

Although the idea and words for a computerized neural network come from the biological ideal of the human brain, an artificial neural network is not a model of a brain, nor does it "learn" in the human sense. It is simply very good at finding patterns. In fact, it can be so good that it "overfits" the data, finding patterns that exist only by chance. In that way, nearly all methods for finding the "best" performing systems share the same problems, whether very simple or complex techniques.

A wide selection of neural network software is available for the personal computer. Many of these can be found through trade magazines such as *Technical Analysis of Stocks & Commodities and Futures*.

**Terminology of Neural Networks**

The brain is composed of cells called neurons, which process and store information. They are unique in the human system because they do not die, which is why we are able to remember. Neurons function in groups called networks, which have thousands of interconnected neurons, and those networks are connected to other neural networks.

Information is received through dendrites and goes directly into a neuron. The data can be passed to other neurons through an output connector called an axon. As the information passes from one neuron to another neuron or neural network, it may pass through a synapse, which can inhibit or enhance the importance of the data going in different directions. A synapse may also be considered a "selector." Figure 9-4 shows a biological neural network and its components.

The human neural network is remarkable in its ability to receive vast
amounts of data, store it away, and make you aware of only the most important items. This sophisticated selection process can vary for each individual and situation. For example, you may no longer hear the ticking of a clock in your bedroom but are instantly aware of the smallest unusual sound coming from a child’s room. We actually hear the clock, but the sound becomes routine and our neural networks do not alert us to action. When the clock stops, an efficient human system will notice.

**Artificial Neural Networks**

Using the same structure as the biological neural network in Figure 9-4, we can show how economic and price information passes through a computerized, artificial, neural network (ANN) to produce a decision on the direction of stock prices.

The first step is shown in Figure 9-5. The system receives a wide assortment of input data, as does the human system. It must select which of these are relevant in finding the answer and assign weighting factors to represent their value. Because “Domestic Health” influences interest rates and subsequently stock prices, we select how each of the five inputs will be used. Two are discarded as irrelevant by giving a weighting factor of zero to their importance; however, all items remain stored in “neurons” for later use.

Unemployment, GDP, and inventories are all considered to contribute to Domestic Health. The value placed on their importance will be decided by the computer neural net program. We can expect the weighting factors for GDP and inventories to be positive, because rising GDP and inventories indicate strong economic activity. Inventories are
Figure 9-5. A three-layer artificial neural network. Economic and price data are received and stored in Level 1 neurons. Information is passed to Level 2 through synapses that reduce or increase the importance of each item, forming new composite values representing the economic health of the country and the anticipated direction of the U.S. dollar. These values are then combined into Level 3, an expected movement in U.S. stocks. In most artificial neural networks each neuron in Level 1 is connected to all neurons in Level 2.

not as clear and get a smaller value. Unemployment gets a negative factor. When unemployment rises, domestic health declines.

The Three-Layer System

Domestic health is only one element needed to forecast stock market prices. Current and anticipated interest rates may be the greatest factor. But interest rates are used to achieve an economic growth target, which may be measured by domestic health. Therefore, each feeds on the other. To make matters more complicated, world political events cause money to move to safety. If Eastern Europeans are buying the U.S. dollar regardless of current interest rates, then rates can go lower. On the other hand, if the United States needs to attract foreign investors to support its debt, then U.S. interest rates must rise to attract buyers, regardless of domestic health.

Each neuron in Level 2 and 3 depends on the weighting factors of each of the neurons that feed it. Unlike the expert system, it is not told the effect
Figure 9-6. Learning by feedback. A neural network program uses the known answers as feedback to find the weighting factors that work.

of each piece of input data, but determines its use by comparing historic examples. For example, the value assigned to domestic health in Figure 9-6 is the result of many inputs, each with unique weighting factors.

The Training Process

The ANN arrives at an answer through a computer-intensive process of pattern recognition. The most popular method is called a genetic algorithm, because random selection in determining the importance of each input causes one solution to be better than another. This “mutation” is used in the same way that natural selection would allow a better species to survive.

The neural network uses the genetic algorithm to “learn” how to arrive at the best answer in a feedback process called training. This method compares the random use of input data and indicators with a known answer until it finds the combination that comes closest to being correct. Figure 9-6 shows the “feedback loop” that mutates weighting factors, using random numbers, until the answers match a large sample of historic situations.

Because this is a trial-and-error process, rather than an analytic approach, the best results could be caused by a coincidental occurrence
of data, rather than cause and effect. Keep in mind the famous warning, *Post hoc ergo propter hoc* (literally, "After this, therefore because of this"), which refers to the error in thinking which assumes that because one event followed another, the second event was caused by the first.

**A Training Example**

We would like to train a neural net to tell us

Should we buy or sell the stock market?

using only the five inputs shown in Table 9-2. To make it simpler:

1. Each input is given an adjusted value, from +100 to −100, indicating whether the current state is strong or weak, high or low, or neutral.
2. Buy signals are given when the total value is above 120; sell signals occur when the value is below −120.
3. Any value from +120 to −120 is considered neutral.

The initial test has only two training cases, that is, the computer is given two sets of input data and the correct answers. History tells us that, when evaluated correctly, Case 1 will give a strong signal and Case 2 a weak signal. To begin, the weighting factors are all set to 1.0. As seen in Table 9-2(a), both Case 1 and Case 2 produce neutral results, which are wrong.

**Trial and Error**

Through the process of random assignment of weighting factors (which was chosen as the "genetic algorithm"), the network “mutates” the pattern. When the ANN arrives at a wrong answer, by comparing it against the historic facts, an error signal is produced and the neural net is told to try again. This is the feedback process. The system then changes the weighting factors until it stumbles on the reverse sign for interest rate and unemployment values (minus instead of plus). The neural net had started by assuming that “low” meant negative and “high” meant positive. But the effect of low or high interest rates is the reverse in the stock market. By changing the weighting factor to −1.0, a correct answer is generated in Table 9-2(b).

Because there were only two cases, many combinations of weighting factors would have given the correct answers. For example, interest rates could have been assigned a factor of −5.0 while all other inputs were given a value of zero. These ambiguities are quickly removed when the number of training cases increases.
Table 9-2. Two training cases

(a) First Test with Unit Weights

<table>
<thead>
<tr>
<th>Input</th>
<th>Case 1</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relative</td>
<td>Value</td>
</tr>
<tr>
<td>GNP</td>
<td>Strong</td>
<td>50</td>
</tr>
<tr>
<td>Unemployment</td>
<td>Low</td>
<td>−25</td>
</tr>
<tr>
<td>Inventories</td>
<td>Low</td>
<td>−50</td>
</tr>
<tr>
<td>U.S. dollar</td>
<td>Vry Strg</td>
<td>75</td>
</tr>
<tr>
<td>Interest rates</td>
<td>Falling</td>
<td>−25</td>
</tr>
<tr>
<td>Total of all test values</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>Buy/sell threshold levels</td>
<td>±125</td>
<td></td>
</tr>
<tr>
<td>Computer training answer</td>
<td>Neutral</td>
<td></td>
</tr>
<tr>
<td>Actual answer should be</td>
<td>Strong</td>
<td></td>
</tr>
</tbody>
</table>

(b) Training Cases with Mutated Weighting Factors

<table>
<thead>
<tr>
<th>Input</th>
<th>Case 1</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relative</td>
<td>Value</td>
</tr>
<tr>
<td>GNP</td>
<td>Strong</td>
<td>50</td>
</tr>
<tr>
<td>Unemployment</td>
<td>Low</td>
<td>−25</td>
</tr>
<tr>
<td>Inventories</td>
<td>Low</td>
<td>−50</td>
</tr>
<tr>
<td>U.S. dollar</td>
<td>Vry Strg</td>
<td>75</td>
</tr>
<tr>
<td>Interest rates</td>
<td>Falling</td>
<td>−25</td>
</tr>
<tr>
<td>Total of all values</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>Buy/sell threshold levels</td>
<td>±125</td>
<td></td>
</tr>
<tr>
<td>Computer training answer</td>
<td>Up signal</td>
<td></td>
</tr>
<tr>
<td>Actual answer should be</td>
<td>Strong</td>
<td></td>
</tr>
</tbody>
</table>

Relative values assigned in a range from −100 to 100. WGT is the weighting factor, set to 1.0 to start. Actual is the response you want to get, or the actual price change over the next 10 days.

Specifying a Neural Network Test

Training the neural network can be a very long process. Allowing the computer to assign weighting factors to countless data items, and continually comparing the answers with the correct one, can take longer than we are prepared to wait. To control the process, it is necessary to put limits on the training and help it along.

Preprocessing. Rather than giving the system all the data possible, select the most significant information. Rather than using automobile and department store sales separately, use a single retail sales figure. Eliminate similar items; each piece of information will continue to be analyzed by the computer over and over again without distinguishing whether two items are the same. Combine some items into indicators and eliminate the less
important elements as redundant. If an index is more complicated than a simple weighting of its elements, then the neural network will not include it properly. Include a trend of prices; the ANN cannot create one itself.

**Break the Problem into Clear Steps.** Solving the problem in a single step may be overly complicated. More important, it becomes very difficult to validate the results. For example, separate forecasts into components. Before looking at the expected move in the stock market, forecast interest rates one week out, or forecast a cut or rise in the prime rate. If the stock market direction is dependent on interest rates, verifying the decision process for rate changes should be a necessary step.

**Choose the Number of Decision Levels.** Two or more inputs are given weights and may be combined into a single item in a new decision level. If all the inputs can be taken two at a time, combined, and then used with the combination of two other inputs, the computer creates an excessive number of "hidden layers." More hidden layers allow more combinations and increase processing time. They allow the solution to be more specific and require much more data to offset possible overfitting. A four-layer system will also take much longer to process than a three-layer one, therefore the three-layer is highly recommended.

**Choosing the Smallest Number of Neurons.** Although limiting the number of "hidden" layers will make the solution faster and more general, the number of neurons that hold intermediate results also can be specified. Just as fewer layers create a more general and faster solution, a small number of neurons in each layer has the same effect. Fewer neurons mean a more general solution.

**Trade-Offs.** As with conventional optimization, neural networks can produce a result that is overfit. Too much data, much of it irrelevant, and too many hidden layers and neurons allow the computer to find spurious patterns. Too few data items, levels, and neurons may make the result so general that it is useless. The analyst must find the proper compromise.

**Forever Learning**

The neural network responds to combinations of events in the manner in which it was "taught." A drop in interest rates without the associated poor economic news (e.g., a flight to the safety of the U.S. dollar) results in a signal to buy equities. The neural network learns that this situation is still good for stocks. But one day, interest rates drop sharply
when investors move their money fearing a plunge in stock prices. You are a buyer of stocks because the move to lower interest rates satisfies the rules. The trade is a large loss as more stock is liquidated. The neural network adds a rule to bypass trades that begin under high volatility conditions.

The system continues to learn. There is no way to know how many different situations will be added to the list of conditions that build a complete network. A synthetic neural network is a technical achievement of large proportions. It can find patterns that cannot be identified by conventional methods (such as multiple regression used in econometrics). It can train itself to determine the importance of each input. But there are many problems it cannot resolve. It will recognize only those inputs that it expects and may not respond properly to combinations of inputs that it has not “seen” before. If there are too many inputs, the network may respond correctly, but to the wrong events. It is the best so far, but it does not guarantee the results within the limits of your investment.

**Fuzzy Logic**

_Fuzzy logic_ is not a brand name, or a description of scatterbrain thinking; it is a formal area of mathematics. Along with neural nets, fuzzy logic pushes the bounds of science. The idea of “fuzziness” describes the lack of precision in normal human conversation and thought. The concept will allow human uncertainty to be introduced to artificial intelligence methods. Think about most casual conversations:

“There were a lot of people at the game.”

“Most of them were tall.”

“It was really cold last night.”

“The market was strong yesterday.”

“The dollar collapsed when the trade deficit was higher than expected.”

Although these conversations do not include specific numbers, we accept and understand what the other person is saying. In fuzzy logic, all is not true or false, 0 or 1, there or not there. It will answer questions such as “If a half-eaten apple is still an apple, how much do you have to eat before it stops being an apple?”

The fuzziness concept includes _fuzzy numbers_, such as “small,” “about 8,” “close to 5,” and “much larger than 10,” as well as _fuzzy quantifiers_, such as “almost,” “several,” and “most.” The phrase “Surprise government reports cause big moves” is a common fuzzy expression.
Fuzzy Reasoning

"Fuzzy events" and "fuzzy statistics" are combined into fuzzy reasoning. Surprisingly, answers to the following examples are remarkably clear to a brain, but not to a machine:

Example 1:  X is a small price move.
              Y is much smaller than X.
              How small is Y?

Example 2:  Most price moves are small.
             Most small price moves are up.
             How many price moves are up?

Example 3:  It is not quite true that the quarterly earnings were very bad.
             It is not true that quarterly earnings were good.
             How bad were the quarterly earnings?*

Practical Solutions

Fuzziness is not intended to describe the same concepts that can be explained precisely by probability (referred to a crisp logic.) Because fuzzy logic and possibility theory are new, mathematicians believe that practical applications will use both (fuzzy) possibilities and (crisp) probabilities. Up to now, the fuzzy part has been assigned ranges to represent commonly used values. For example, in expressing the S&P price change, we might have the following:

<table>
<thead>
<tr>
<th>Description of Change</th>
<th>Value of an S&amp;P Rise or Fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>unchanged</td>
<td>± 20 points</td>
</tr>
<tr>
<td>small</td>
<td>± 145 points</td>
</tr>
<tr>
<td>medium/normal</td>
<td>± 150 to 400 points</td>
</tr>
<tr>
<td>large</td>
<td>more than ± 400 points</td>
</tr>
</tbody>
</table>

*The answers to the examples are very small (1) and small (2).
The advantage of being human is that we can express these ranges in a vague but sophisticated way. We have defined “unchanged” to be a close within 20 points of the previous close; yet following a few very volatile days, we might call a move of ± 50 points “unchanged.” We all seem to have the same understanding of relative volatility when we speak to one another, but trying to put it on paper evokes disagreement.

**Suboptimization or Deoptimisation**

Fuzzy logic may bring a better solution than other methods that are now being used to develop trading models. By its very nature, a fuzzy solution must be general. Writing fuzzy rules for a trading program will not have to be as precise as traditional specifications. That also means there should be fewer rules and a more robust solution. No matter how hard we try, it may be impossible to overfit the solution using fuzzy data.

**State of the Art**

Fuzzy systems have been combined with neural networks and expert systems, which provide a framework for “learning.” Neural nets provide the behavioral structure so that correct answers are reinforced and incorrect ones are rejected. Expert systems give the program a knowledge base.

Japanese firms have led the financial industry in the application of fuzzy expert systems. It is said that programs already exist for financial dealing, especially stock market trading. These models have been based entirely on price information, but may soon include expectations, or “feelings,” about political outcome. It seems that, once the idea is planted, the technology moves forward at a furious pace.
PART 3

Making a Trading Strategy Robust
10

Testing for Robustness

A trading strategy is robust if it is successful under many different conditions. It is especially good if it works under situations that are very different from those used in testing, for example, a more volatile move to new high prices.

Many users can blame the speed of the computer for trading systems that do not work. Combined with strategy-testing and statistical software, the computer has made it too easy to simulate thousands of trading rules and techniques. Preprogrammed strategies, countless indicators, and the ability to create your own variations often draw inexperienced users into an indiscriminating and unfocused approach to testing. In the end, the computer has tested too much and used too little as criteria for success. Often, the resulting trading programs appear to be remarkably profitable but in reality are complete failures.

Overfitting

A system that has been tailored to work on a specific period of historic data is called overfit. Everyone who develops a trading system will use past data to verify the results. It would be irresponsible to define a set of trading rules, open an account, and begin trading without knowing whether those rules would have worked in the past. The historic risk will give you an indication of the investment size needed to achieve your goal and survive the interim losses required to reach your objective.

A careful study of historic results will often point out an area of high risk. Sometimes a simple rule is all that is needed to reduce the risk to a comfortable level. For example,
Reduce the size of the entry position as the market increases in volatility. Further analysis could lead to other rules:

Close out all positions if the S&P drops more than 1500 points in 3 days.

or,

Close out all long positions on Friday if the S&P has dropped more than 10% during the previous week.

These rules move from a general, logical risk control to very specific guidance intended to isolate one or two past problems. Where do these changes stop being reasonable and start being manipulative? The answer is rarely clear.

In this chapter, we will consider a system robust if it does not depend on a narrow set of conditions. A profitable 10-day moving average system will not be used if similar 8-day and 12-day systems generate losses. The best system is one that is profitable for a broad set of parameters, including trend speeds, risk control, profit-taking, and filters. The trader gains a more dependable program when nearly any choice of parameters is likely to give profits.

**Separating Robustness from Parameter Selection**

There is a clear separation between determining that a trading strategy is valid, and being able to use that method to produce profits in the future. Historic testing can verify a premise and show which sets of parameters, or variables, were successful in the past. But this does not mean the parameters that generated the most profits in the past will lead to future profits. And, when there are two tests with similar historic success, which will be the best?

Comprehensive testing, called *optimization*, results in some cases that show profits and others that have losses. The profits show that the logic behind the strategy is sound. The more cases that are profitable, the more confidence we have in the trading method. However, the parameters that gave profits in historic tests do not always generate profits during real trading. The ability to choose, in advance, which parameters will give future profits is a separate problem from defining robust trading rules.

Most of this chapter will focus on how to build a robust trading system; the more robust it is, the less it will depend on picking the right parameters. We can assume that an arbitrary choice of parameters will yield the
average performance; therefore, we should be sure that the average is good.

Principles

The development of a trading model that is independent of parameter selection is always the ideal solution. But it is elusive. It is the opposite concept of an arbitrage, which takes advantage of distinct economic anomalies. This chapter will set down a procedure that will greatly improve the robustness of any trading model. It is not a simple process, but it can be implemented one part at a time. The more that is done, the better the results.

No model is without some limitations or restrictions. Each has a purpose that requires some definitions of its operating environment. It is certainly reasonable to exclude a 3-day moving average from a long-term trading system. Both trading frequency and risk can be narrowed to ranges that make sense for the strategy as well as for the type of business. Exploiting a price pattern is valid if the patterns can be identified in advance. It is within this framework that the program can be robust.

Some basic principles of parameter selection assure a choice that yields superior results. These include slower trends and fewer artificial risk controls. This chapter will draw on previous conclusions to suggest a set of rules and a testing approach that will give test results similar to real trading.

Example of Optimized Performance

Table 10-1 compares test results of a moving average system with a percentage stop-loss. The speed of the moving average (from 5 to 50 days) and the percentage stop-loss (from 0 percent to 2.0 percent) are the two parameters needed to trade. A TeleTrac optimization was used to find the returns for each combination of speed and stop for the Hang Seng Index during each of the calendar years 1991 and 1992.

If we had used the best results of the 1-year test on 1991 data to select the parameters to be traded in 1992, we would have tended toward the slower moving averages. The highest return of 17.4 percent was given by the 45-day trend. Speeds from 5 to 30 days showed erratic results and only a few small profits.

The 1992 results show nearly the opposite. Moving averages from 5 to 20 days had large returns, while speeds from 40 to 50 days had the worst results. Had we chosen the parameters that posted the highest profits in 1991, the 1992 performance would have been only 1.3 percent (if executed perfectly).
Table 10-1. Moving Average Optimization of the Hang Seng Index

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Moving Avg Days</th>
<th>% Stop-Loss</th>
<th>1991 P/L</th>
<th>1992 P/L</th>
<th>Change in P/L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>1.0</td>
<td>1.6</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.0</td>
<td>0.1</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1.0</td>
<td>11.3</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1.0</td>
<td>3.4</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1.0</td>
<td>3.7</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1.0</td>
<td>13.3</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1.0</td>
<td>1.0</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1.0</td>
<td>13.0</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>1.0</td>
<td>17.4</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1.0</td>
<td>3.7</td>
<td>2.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

This simple moving average test shows the typical inconsistency in the performance of the “best” choices when a test uses only a small amount of data. The area of highest profits in 1991 produced the worst results in 1992.

The Underlying Method for Determining Robustness

We can improve the selection of parameters by focusing on systems that have the broadest success. If all test cases are profitable, we would have the perfect robust system and any selection of parameters should return profits.
To measure which strategies are better than others, we will define test procedures that measure the results using the *average and standard deviation of all tests*. The highest average alone is not enough. A smaller standard deviation shows the consistency of performance for all tests. A *Best Choice Index* combines both values by subtracting the standard deviation from the average:

$$\text{Best Choice Index} = \text{Average returns} - 1 \text{ Standard deviation of returns}$$

Because one standard deviation represents a grouping of 68 percent of all data, the Best Choice Index tells us that this system gives an 84 percent chance of a return greater than or equal to the Best Choice Index value. Remember that losses are half the probability, on the left part of the distribution curve. For example, if all combinations of test returns averaged a rate of return (ROR) of 14 percent with a standard deviation of 6 percent, we have

- An 84 percent chance that any choice will yield returns greater than 8 percent (the average minus 1 standard deviation).
- A 97.5 percent chance that any choice will yield returns greater than 2 percent (the average minus 2 standard deviations).
- A 99.5 percent chance that any choice will yield returns greater than −4 percent (the average minus 3 standard deviations).

The minimum test criterion should be an 84 percent chance of success, given by the Best Choice Index.

**Testing Process**

The total solution is the *test process*. It begins with conceptualization. It must be followed by clear steps that lead to a well-defined result. Experience shows that if you do not control the process, the process will control you. The test procedure can be separated into five parts:

1. Deciding what to test.
2. Deciding how to test it.
3. Evaluating the results.
4. Choosing the specific parameters to trade.
5. Trading and monitoring performance.

Each of these steps is critical to the success of the program. Setting up this process for the first time will take a lot of careful work, but most of
it will only need to be done once. There will be many decisions to make with regard to the data, testing software, and the method of evaluation. Because the proper development of a trading strategy is so important to its success, these issues will be discussed in detail in this chapter. Box 10-1 provides a checklist that will serve as a reminder.

Part 1: Deciding What to Test

Before you begin testing, define the system and the test plan completely. You must tell the computer what to do, not allow the computer to tell you. Do not drift from one idea to another as you reach obstacles. Try to follow the original idea to completion and learn its advantages and disadvantages.

Step 1. Is the Strategy Logical?

Did you write the rules before you began testing? Where did you get your ideas? Successful trading programs are based on sound ideas such as economic relationships (e.g., arbitrage, seasonality, and the spreading of strong and weak economies) or valid technical strategies (e.g., breakout of a support or resistance level, selling volatility with options, or use of divergence). Letting the computer uncover an obscure short-term pattern, no matter how reliable it seems, is not a sound trading approach. Price patterns can always be found, but they have doubtful predictability and often change without notice.

When you develop your program, the strategy must make sense for the market and fit your own objectives, as in the following examples:

- For the stock market, you might want a long-term buying strategy with no short positions.
- For the bond market, a long-term strategy that parallels slow-changing economic and government policy might be most conservative.
- For foreign exchange, a short-term method that would buy or sell in the direction of an intraday breakout with small profit-taking objectives might make more sense for tactical hedging and limited overnight positions.

Using Logical Ideas. A logical idea does not need to be based on fundamentals. Years of watching price movement on Chicago’s International Monetary Market (IMM) may give you the idea that dependable entry signals occur only during the three periods of high volume each day—at the
Box 10-1. CHECKLIST FOR ROBUST TESTING

PART 1: Deciding What to Test
☑ 1. Is the strategy logical?
☑ 2. Can you program all the rules?
☑ 3. Does the strategy make sense only under certain conditions?
☑ 4. Take a guess as to the expected results.

PART 2: Deciding How to Test
☑ 5. Choose the testing tools and method.
☑ 6. Do you have enough of the “right” data?
☑ 7. Have you included realistic transaction costs?
☑ 8. Will you test a full range of parameters?
☑ 9. In what order will the parameters be tested?
☐ 10. Are the parameters distributed properly?
☐ 11. Have you defined the evaluation criteria?
☐ 12. How will the output be presented?

PART 3: Evaluating the Results
☑ 13. Are the calculations correct?
☑ 14. Were there enough trades to be “significant?”
☑ 15. Does the trading system produce profits for most combinations of parameters?
☐ 16. Did logic changes improve overall test performance?
☐ 17. How did it perform on out-of-sample data?

PART 4: Choosing the Specific Parameters to Trade
☐ 18. Did the last test include the most recent data?
☐ 19. Did you choose from an area of broad success?
☐ 20. Are profits distributed relatively evenly over the tested history?
☐ 21. Are the profits per trade large enough to absorb errors?
☐ 22. Did the historic results show any large losses due to price shocks?
☐ 23. Have you risk-adjusted the returns to your acceptable risk level?

PART 5: Trading and Monitoring Performance
☐ 24. Are you following the same rules that were tested?
☐ 25. Are you trading the same data that was tested?
☐ 26. Are you monitoring the difference between the system and actual entries and exits?
open, close, and just after traders return from lunch. The low volume periods between give less dependable indication of direction and require a more demanding price move to enter a trade. The important rule is to know what you want to do, and then use the computer to verify your idea. You must control the process.

**Starting with One Idea and Ending with Another.** Be sure that computer feedback does not cause you to stray from your original idea. A logical strategy can evolve into meaningless patterns. There is a natural tendency to explain why a system must be fundamentally sound, simply because you have already seen that the test results are good.

**Step 2. Can You Program All the Rules?**

*Can all the rules in the trading strategy be entered into the computer or a spreadsheet program? Have you assumed anything that was not programmed?* A strategy that cannot be tested cannot be evaluated. If you assume that you would not have been caught in a price shock because the program does not trade overnight, then you leave yourself open to unexpected losses, undercapitalization, and justifiable criticism.

Writing clear trading rules is essential to testing. You must be certain that you can account for entry and exit conditions, risk control, types of orders, time of day, and other situations that completely describe your plan. Writing the rules will tell you the type of data needed for testing (whether it is only prices, the Producer Price Index, or API statistics) the frequency and extent of the data (open, high, low, close, or 30-minute prices with tick volume). As carefully as you try, you will always need to add details later.

**Intraday Breakout Example.** Start with the most basic approach, omitting risk control, profit-taking, or qualified entries. If you believe that an intraday breakout system is a sound idea, then first test only the breakout entry and the basic exit signals. You might want to close out the trade at the end of each day; or, you might exit if prices reverse and breakout in the opposite direction. It is important that you know whether the underlying idea works before adding profit-taking, risk control, and other more specialized features.

Decide which parts of the system can vary. You know that a breakout early in the day allows more time to reach bigger profits during the rest of the trading session. Therefore, you will want to test the time of the breakout. You will not want to accept an entry signal late in the day, because of the limited potential for profits before the close.
If the program is fully computerized, you will want to look at the data no more often than every 5 minutes. Although you may be able to execute an order within 60 seconds of a breakout, it is not practical to assume good executions. Using 5-minute bars for testing; rather than 1 minute, will also reduce the time needed to test the strategy.

Before you start testing, you know that an intraday breakout system depends on the period over which the breakout is measured, the time of day for entry, the size of the profit-taking objective, and some risk control.

**Trend System Example.** All systems have common features: entry and exit rules, risk control, and possibly profit-taking situations. A trend system requires a trend speed. This can vary significantly with your application and objectives. Equities programs, with little leverage and higher transaction costs, require a range from 50 to 500 days. A futures trader, with margins of only 5%, will favor faster trends, from 5 to 30 days.

It is a mistake to use a smoothing approach on intraday data. As the time period between data observations gets shorter, the level of noise increases. Because of illiquid periods in all markets, prices can jump in either direction without indicating a true trend change. This causes frequent false signals that cannot be eliminated by using a longer trend based on the same intraday prices. The combination of intraday noise and trend lag will be a difficult obstacle to overcome.

**Step 3. Does the Strategy Make Sense Only under Certain Conditions?**

Decide, in advance, whether the strategy targets certain market movement, or a specific set of conditions. The idea may only make sense for long or short time intervals. For example, a day-trading program using 15-minute data would not use a 200-period moving average, while a long-term investment program in stocks would not use a 3-day trend. By defining the range over which the trading model will operate, you reduce the chance of being diverted from your objective. Write out the most reasonable test range for each of the parameters that are considered important to the strategy. The more you can define your expectations, the better the results.

**Step 4. Take a Guess as to the Expected Results**

Decide the expected rate of return, the percentage of profitable trades, and the size of the losses. The objective is to compare the test results with your
expectations. Whether the results are much better or worse than planned, when you have a basis for evaluation it will be easier to correct and move forward with the development of the system. To say "something is wrong" with the test results, you must first decide what you expect.

Part 2: Deciding
How to Test

Step 5. Choose the Testing Tools and Method

With more sophisticated strategy-testing software, it is no longer necessary to program the trading method in FORTRAN, BASIC, or C to test its success. In a few minutes, using a strategy testing package such as TeleTrac, Omega's System Writer, or even a Lotus or Quattro spreadsheet program, you can have a good idea of the viability of the technique.

An increasing number of programmable graphics terminals and new strategy-testing software are available at very competitive prices. They all have the advantage of calculating profits and losses accurately, the flexibility of rule changes and data selection, and the ability to plot both data and profitability. In some cases, results can be read into spreadsheets for further evaluation. The time saved is well worth the price. For the more sophisticated analysts, supplementary software such as Manugistics Statgraphics and Mathsoft Mathcad are impressive tools for evaluating complex statistical relationships and expressing mathematical formulas.

Long Test, Short Test, or "Step-Forward" Test? The pattern in Table 10-1 is not unusual. Tests using a small amount of data give results showing that many combinations of parameters will work. The shorter the test period, the more profitable the system will appear. Consider a bond market that has moved steadily up for 3 months. If there were only small retracements, then any moving average from 10 days and longer would have yielded the same results, which is the net move from the beginning to the end of the period (see Figure 10-1(a)).

When a short test interval has one or more price swings, the slower trends give back profits, while some of the faster ones are very successful. The size of the swing and the amount of market noise determine which trend speeds are best (see Figure 10-1(b)). In general, tests of small amounts of data give:

- Individual and average test results that are much higher
- Risk that is sometimes lower
- Profitable results for more models that trade faster
- Erratic forecasting ability
Figure 10-1. Moving average results for a short test period. (a) A short period with a strongly trending market allows most trend speeds to perform well. (b) A short period with price swings may allow fast trends to capture profits, but net losses for the longer term trends.

It is much more difficult to find a trading method that is good over longer test periods. The best tested performance (annualized rate of return) of a system tested over many years will never be as high as the rate of return of a similar system tested over a few months or a year. Using more data, you should expect:
- Much lower returns
- Larger risk when positions are held longer
- Difficulty in getting consistent profits from short-term trading
- Questions as to the relevance of older data
- Better forecasting ability

Therefore, test results using smaller amounts of data look better, but do not perform as expected; results based on longer tests look worse but perform closer to expectations. You should not be disappointed in the results of a long test period when compared with shorter tests. It is only that the shorter tests are misleading.

**Select a Long, Representative Test Period.** When more data are tested, there is a greater variety of unusual situations, longer profitable price moves, sequences of losses, and price shocks. When longer periods are tested, both risk and reward increase; however, risk increases faster than returns. Testing shorter periods can give an unrealistically small risk, cause undercapitalized trading and fatal results. A good rule is to be certain that the data contain *two full cycles*, that is, there should be two clear bull markets, two bear markets and two prolonged sideways intervals.

Because results never look as good when the same strategy is tested over longer periods, you might argue that markets have changed and the old data are no longer representative; that globalization and regional alliances have changed the price relationships and patterns in many sectors, or that government controls will prevent an economic collapse. By saying that the market will continue to exhibit only the price patterns seen recently is unrealistic. It will evolve to new patterns; however, we have no way of knowing what they will be. The past contains the most accessible, practical, and realistic examples of changing situations. Box 10-2 shows that performance drops but predictability increases with the use of more data. Short test periods produce unreasonable expectations of profits.

**"Step-Forward Testing" versus One Long Test.** The technique of "step-forward testing" seems to be a sensible approach to resolving some of the testing dilemmas. It works as follows:

1. Select a short data interval, called a "test window" (e.g., 2 years of data).
2. Test (optimize) a full set of parameters on the test window and select the "best."
3. Run the model on a short period of out-of-sample data, immediately following the test window (e.g., the next 3 months), using the “best” parameters (see Figure 10-2).

4. Collect performance data on the “out-of-sample” period, including a comparison with the “in-sample” test-window returns.

5. Move the test window forward and repeat steps 2 through 4 until done.

6. The parameters that perform most consistently in the out-of-sample period are considered best.

**Hidden Problems.** Step-forward testing seems to duplicate the way we would operate a trading program. But there are hidden problems:

- **Shorter test periods favor faster strategies that produce higher profits and more trades.** But the performance of fast-trading strategies varies sharply from one test window to another, as discussed in the previous section and shown in Table 10-2. The “best” parameters are not a good choice for trading.

- **Short test periods do not represent long-term trading fairly.** Each short test period can have only a few long-term trades, and they may start and end in the middle of a trade. Moving the test window forward does not correct the treatment of these trades.

- **Retesting the same system with modified rules means that the “out-of-sample” data is no longer new.** This is called “feedback.” Once used, the data is no longer “out-of-sample.” You know what to expect and how to make the data patterns show more profits.

The step-forward process will usually select an inconsistent, fast-trading method over a better long-term system simply because the test window forces this result. Instead, use all the data in one long test to get continuous performance over as many changing patterns as possible.

**Step 6: Do You Have Enough of the “Right” Data?**

The more data you test, the more situations the program will experience. There must be at least two bull markets, two bear markets, and two sideways periods. Unless you can prove that the older data is misleading, or no longer valid, you should use as much data as possible. Put some data aside for out-of-sample validation after the final system has been selected. This is discussed further in Step 18.
Box 10-2. MORE DATA IMPROVE TEST RESULTS

Using more data produces more consistent and realistic results. Final results may show that risk is higher and profits are lower, but these figures are more likely to be achieved in trading. It is more difficult to find persistent short-term patterns in a longer data series; therefore, selections favor slower trading. Long-term solutions, in turn, include realistic equity fluctuations because they cannot be fine-tuned to avoid specific losing periods. This performance profile shows higher risk and makes it necessary to have higher capitalization.

A simple test of the MATIF CAC-40 Index (Table 10-2) shows the predictive ability of tests based on 1, 2, 3, and 5 years of data. The system tested was

1. An exponential moving average from 5 to 50 days, in increments of 5 days.
2. A trend change criterion ("filter") from 0 to 10 points, in increments of 1 point.
3. A buy signal that occurred when the trend turned up by the amount of the filter; a sell signal that was generated when the trend turned down by the amount of the filter value.

The highest profits for each test determined the trend speed and entry filter that were to be used to evaluate the next 1 year of data. The averages for each test case were compared.

<table>
<thead>
<tr>
<th>Table 10-2. Tests of CAC-40 (MATIF) French Stock Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Year Test</td>
</tr>
<tr>
<td>Year(s) Tested</td>
</tr>
<tr>
<td>1982</td>
</tr>
<tr>
<td>1983</td>
</tr>
<tr>
<td>1984</td>
</tr>
<tr>
<td>1985</td>
</tr>
<tr>
<td>1986</td>
</tr>
<tr>
<td>1987</td>
</tr>
<tr>
<td>1988</td>
</tr>
<tr>
<td>1989</td>
</tr>
<tr>
<td>1990</td>
</tr>
<tr>
<td>1991</td>
</tr>
<tr>
<td>1992</td>
</tr>
<tr>
<td>Avg/Yr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2-Year Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year(s) Tested</td>
</tr>
<tr>
<td>82-83</td>
</tr>
<tr>
<td>83-84</td>
</tr>
<tr>
<td>84-85</td>
</tr>
<tr>
<td>85-86</td>
</tr>
<tr>
<td>86-87</td>
</tr>
<tr>
<td>87-88</td>
</tr>
<tr>
<td>88-89</td>
</tr>
<tr>
<td>89-90</td>
</tr>
<tr>
<td>90-91</td>
</tr>
<tr>
<td>91-92</td>
</tr>
<tr>
<td>Avg/Yr</td>
</tr>
</tbody>
</table>

Speed and filter vary.  
Test is sensitive to current patterns.  
Average trend slows.  
In-sample profits decline.  
1 year ahead improves.
Summary of results

- **1-year test.** The best trend speed and filter varied considerably from year to year. The average speed was lowest of all tests, showing that a fast trend often looks best for a short test period. Performance was inconsistent in 1-year-ahead tests, averaging a loss of 41 points. Using 1 year of data to forecast 1 year ahead does not look promising.

- **2-year test.** Overall profits per year declined and the average best choice slowed slightly, showing that more test data become harder to fit. However, results for 1 year ahead increased and became net positive.

- **3-year test.** The performance pattern continued to improve. The 1-year-ahead tests for 1988 and 1989, which show large losses, were studied to find out that those years posted new highs. Performance in the out-of-sample data seems to be better when the price movement is within the range seen in the tested period.

- **5-year test.** Improvement continued overall. Longer trends were selected and tested performance declined. These are especially good results because 1985 and 1986, which showed large profits in the 3-year test, were not part of this out-of-sample performance.

---

**Table 10.2.** Tests of CAC-40 (Matthew) French Stock Index (Continued)

<table>
<thead>
<tr>
<th>Year(s) Tested</th>
<th>Best Performance</th>
<th>1 Year Ahead</th>
<th>Year(s) Tested</th>
<th>Best Performance</th>
<th>1 Year Ahead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Filter</td>
<td>P/L</td>
<td>Speed</td>
<td>Filter</td>
<td>P/L</td>
</tr>
<tr>
<td>82-84</td>
<td>15</td>
<td>2</td>
<td>162</td>
<td>215</td>
<td></td>
</tr>
<tr>
<td>83-85</td>
<td>20</td>
<td>2</td>
<td>472</td>
<td>119</td>
<td></td>
</tr>
<tr>
<td>84-86</td>
<td>20</td>
<td>3</td>
<td>1165</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td>84-87</td>
<td>20</td>
<td>3</td>
<td>1414</td>
<td>(454)</td>
<td></td>
</tr>
<tr>
<td>86-88</td>
<td>25</td>
<td>9</td>
<td>1225</td>
<td>(388)</td>
<td></td>
</tr>
<tr>
<td>87-89</td>
<td>50</td>
<td>1</td>
<td>910</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td>88-90</td>
<td>10</td>
<td>7</td>
<td>900</td>
<td>(71)</td>
<td></td>
</tr>
<tr>
<td>89-91</td>
<td>30</td>
<td>4</td>
<td>832</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>90-92</td>
<td>25</td>
<td>10</td>
<td>1072</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Avg/Yr</td>
<td>24</td>
<td>5</td>
<td>302</td>
<td>27</td>
<td></td>
</tr>
</tbody>
</table>

Average trend is slower.
In-sample profits are lower.
1 year ahead is better.

Average trend is slower.
In-sample profits are lower.
1 year ahead is best.
New highs in 1988 generated losses because it is not part of sample data.
Are You Testing the Same Data That You Will Trade? Do not test one set of data, then trade another. Do not use a "continuation" series because either the gaps have been removed, or they cause windfall profits or losses that would not have happened in trading. A "perpetual" contract has prices that never existed and usually dampens any severe price move causing the risks to look smaller.

Did You Verify the Accuracy of the Data? Data can be inaccurate even when prepared by a reliable vendor. Look for prices at the beginning or end of a contract that are completely different. Sometimes the data will have prices from another market that have not been erased, or an erroneous date one or two years earlier. Check for blank or zero entries. If you chart the data, you will easily see errors. The ones that are too small to see can be ignored.

Special Cases in Selecting Test Data. It is not always possible to have enough data for testing. New markets or changing situations may render old data questionable. Or, you are looking to profit from a recent price pattern, without expectation of using the system for very long. The following sections offer some alternatives.

Selecting Similar Data Periods. A stock that has dropped to a very low level can have a very different performance pattern from a period of high prices and high volatility. Selecting similar historic periods, such as those following a prolonged decline, or after a sell-off of 10%, may be the only way to model your strategy.
Using Cash Markets to Model Futures. Cash markets are often used to test a system that will be traded as a new futures contract; however, a new contract can be illiquid. A good model will account for similar situations in other new markets, adapting to the change between the cash and futures. Because there are many examples of changing markets, this should be a successful exercise.

Stock and Futures Markets under Special Situations. All markets go through severe changes: a corporate scandal or mismanagement, sudden new competition or government regulation; a price shock in coffee or orange juice due to a freeze. These special situations must be also studied separately, rather than absorbing them into the flow of everyday price movement. Market reaction to special situations is often similar because of the human response, rather than the fundamentals of a company or commodity. Similar cases can be found in other markets. When the special situation is a "price shock," a new set of rules can be used. This is discussed in Chapter 7.

Structural Changes and Not Enough Data. The European Monetary System (EMS) imposed a structure on participating currencies that had little precedent. A previous period, under the Bretton Woods agreement, may not provide enough similarity or adequate data for modeling a trading strategy. In this case, a fundamental analysis is the only course. Results based on small amounts of data are unreliable. A sound understanding of the fundamental interaction and the rules under which the new agreement operates may allow some confirmation by testing. To date, the EMS has proved to be unstable, therefore, a test of the 1 to 2 years of data would have led to poor results.

Creating More Data. For some markets, it is possible to create synthetic data. By studying volatility at different price levels, sequences of runs, variation in periodicity between highs and lows, and seasonality, it is possible to use random price generation to create data with the same statistical qualities as the one being evaluated. Synthetic data gives you the ability to test more situations and develop a more robust solution, but it is not the same as real data. It is best to use synthetic data first, before testing actual data.

Type of Data to Test. The data used for testing strategies should always be the same as the data to be traded. This is very straightforward for stocks, but becomes more difficult when you use foreign exchange or futures prices. The forex market will require adjustment for interest over the holding period, but the futures market presents the greatest problem. Although the nearest futures has the greatest liquidity, it may trade actively for as little as one month, and rarely more than three months. This frequent expiration makes testing inconvenient. The following sections will show how to fix this problem.
Original Data Series. For stocks, foreign exchange, interest rates, or other cash market data, a long series of original, unadjusted data is available for testing. When trading the cash market you will also need the spot interest rate to calculate a forward price. Treating the cash price as a valid entry and exit point omits the need to roll the position forward daily. Each rollover has an implied transaction cost that eats away at profits. Alternately, you can use spot prices for entry and exit, and calculate the net interest rate credit and debit when liquidating the position. Additional transaction costs must be included in testing each time the position is rolled over.

Futures Contracts. Original futures contract data can also be used for testing, without modification. Use the following steps:

1. Read the futures contract data.
2. Start the strategy calculations at the beginning of the series ("winding up the strategy"), or as soon as there is enough data.
3. Begin taking positions in the new contract on a specific date, or on the day that the previous contract stopped trading. If the previous contract had an open position when it was "rolled," then assume the same position on the same day in the new contract. Exceptions should be made for markets that do not have a high correlation between delivery months, such as livestock, or at critical seasonal periods such as the change for old to new crop for grains, or February-March in heating oil.
4. Exit any trade on a specific preset date before expiration. For interest rates, this is usually the last day of the month prior to delivery. For currencies, it is about 3 days before expiration.

This method is inconvenient because results are usually given by contract. For a 10-year test of interest rate futures, 40 separate sets of results must be accumulated. In addition, it is difficult to assess maximum drawdown unless you can treat the segments of data as a continuous equity stream.

Continuous Data Series for Futures. Many data vendors provide a constructed series that pieces together the nearest futures contract to create a new series (e.g., a 3-month price) calculated in a way that resembles the London Metal Exchange forward contracts. These choices are unacceptable for testing because they do not show the data that will be traded in a way that can duplicate a realistic trading environment. The constructed 3-month series, with interpolated carrying charges or interest, is frequently a smoothed version of prices that occurred at that time, reducing both the profit and risk.
**Gap-Adjusted Series and Index Series.** A gap-adjusted data series is a good alternative for most technical applications in futures. It puts the nearest-to-delivery segments together into a single price series by closing the gaps at the time one contract rolls into the next. By proceeding backward, the most recent futures or forward contact has today’s prices, and the older contracts are adjusted up or down according to the gaps.

The gap-adjusted series works well for trend-following applications and strategies where the comparative price, rather than the actual price, is needed. It does not work for chart analysis, economic studies (supply/demand/price relationships), and similar uses. One problem with gap-adjusting, where older prices are changed, is that the very old data can take on negative or unrealistic values. Because the prices are not real, the rate of return and risk measurements must refer back to the actual prices, rather than base their values on the gap-adjusted series.

With one additional step—indexing—the gap-adjusted series becomes more useful. Indexing is simply starting with the value of 100 (or 1000, depending on convenience), then adding or subtracting successive values as a percentage change. For example,

\[
\text{index} = \text{index}[1] + (\text{price} - \text{price}[1])/\text{price}[1]
\]

Today’s new index value, index, is yesterday’s value, index[1], plus the percentage change in yesterday’s price. The notation [1] means the 1-day prior value. The index price represents a percentage change and allows simple comparisons between returns of different markets. It eliminates the need to reference the original price data to calculate risk and returns.

**Building a Gap-Adjusted Series.** If you are working with futures contracts, a continuous series can be very useful. There are three steps to follow, (1) creating a continuous series with duplicate entries on the day of the rollover, (2) gap-adjusting the series, and (3) indexing. Figure 10-3 shows a flowchart of this process beginning with Step 2. Use the prices in Table 10-3 to follow the flowchart. For example, if the S&P 500 were being combined, Step 1 causes the June 93 contract to stop on the last day of May, and the September 93 contract to start on that day. Step 2 would gap-adjust the prices, working backward, whenever it identified a duplicate date. In Table 10-3 the June contract values are adjusted up by 12.00, equal to the roll-forward gap on May 31. Step 3 would assign 100 to the first value, then calculate the percentage changes for each successive entry. Note: A clever analyst can eliminate Step 2 if an index is the only output.

**Alternatives.** The only remaining problem with gap-adjusting is that transaction costs cannot be posted at the time of the roll-forward, because that date can no longer be identified. It may be more difficult, but prefer-
Step 2: Gap-Adjusting

1. Start at the most recent data entry and scan backward. Initialize BIAS, the accumulated size of the gap, and initialize price counter N.

2. If the prior date is not equal to current date, add BIAS and keep data, otherwise add the new gap to BIAS and skip this duplicate entry.

3. If done with data, then go to Step 3, otherwise increment N to look at next older data.

Step 3: Indexing

4. Start at oldest data, M and initialize index to 100.

5. Increment M to look at next sequential data item.

6. Calculate index value as the percentage change.

7. If not done, continue at (5).

Figure 10-3. Flowchart for continuous, gap-adjusted, indexed series. Once you have a continuous series built (Step 1) according to the form shown in Table 10-3, it becomes easy to scan backward and gap-adjust, the go forward to index.

able, to write program logic around the continuation file, which contains the duplicate dates and data. When a duplicate date is encountered, the old trade is closed out and the new trade entered.

Shock-Adjusted Series. A FORTRAN program for removing price shocks, then restoring the continuity of the data by indexing, can be found in Chapter 7. It is a similar program to the one in Figure 10-3 and gives coding details.
Table 10-3. Sample S&P Prices Combined before Gap-Adjusting

<table>
<thead>
<tr>
<th>Contract</th>
<th>Date</th>
<th>Price</th>
<th>Gap-Adj Value</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>JUN93</td>
<td>930528</td>
<td>451.50</td>
<td>463.50</td>
<td>100.000</td>
</tr>
<tr>
<td>JUN93</td>
<td>930529</td>
<td>449.25</td>
<td>461.25</td>
<td>99.500</td>
</tr>
<tr>
<td>JUN93</td>
<td>930530</td>
<td>446.50</td>
<td>458.50</td>
<td>99.388</td>
</tr>
<tr>
<td>JUN93</td>
<td>930531</td>
<td>448.00</td>
<td>no entry</td>
<td>99.225</td>
</tr>
<tr>
<td>SEP93</td>
<td>930531</td>
<td>460.00</td>
<td>460.00</td>
<td>no entry</td>
</tr>
<tr>
<td>SEP93</td>
<td>930601</td>
<td>462.50</td>
<td>462.50</td>
<td>100.005</td>
</tr>
<tr>
<td>SEP93</td>
<td>930604</td>
<td>460.75</td>
<td>460.75</td>
<td>99.622</td>
</tr>
</tbody>
</table>

S&P prices have been combined into a single series, and still show the original prices. A duplicate entry appears on May 31, which will be the date of the rollover where the gap is adjusted. The "Gap-Adj" and "Index" columns show the values after those steps have been completed.

Step 7. Have You Included Realistic Transaction Costs?

Transaction costs include brokerage and slippage. But other factors reduce performance.

**Do You Expect Any Missed Trades?** "Unables" have a great impact on results because they reduce only the profits and not the losses. If you over-trade the liquidity of the market, then unables become an important factor. Programs that trade intraday will face more problems than those that trade on the close. Part of a successful program is achieving actual trading results similar to expectations. A full discussion of slippage and unables can be found in Chapter 2.

Step 8. Will You Test a Full Range of Parameters?

Determine, in advance, the range of parameters that is sensible for this strategy. If you are trading stocks for an institutional portfolio, a moving average test range may be 50 to 400 days. Stop-losses must be equally large. However, do not prescan and remove very fast and slow ranges because they showed losses. That is the same as eliminating everything except the one set of parameters that was profitable. You cannot develop a robust model by looking at a narrow range that has been preselected to work.

Step 9. In What Order Will the Parameters Be Tested?

Test the most important variables first, the ones that cause the largest change in performance. That would be the number of days (the "period")