Information in this document is subject to change without notice.

THE TRADING SYSTEMS IN THIS BOOK ARE EXAMPLES ONLY, AND HAVE BEEN INCLUDED SOLELY FOR EDUCATIONAL PURPOSES. OMEGA RESEARCH DOES NOT RECOMMEND THAT YOU USE ANY SUCH TRADING SYSTEM, AS THE USE OF ANY SUCH TRADING SYSTEM DOES NOT GUARANTEE THAT YOU WILL MAKE PROFITS, INCREASE PROFITS, OR MINIMIZE LOSSES. THE SOLE INTENDED USES OF THE TRADING SYSTEMS INCLUDED IN THIS BOOK ARE TO DEMONSTRATE THE WAYS IN WHICH EASYLANGUAGE CAN BE USED TO DESIGN PERSONAL TRADING SYSTEMS AND TO SHOW SOME EXAMPLES OF HOW CERTAIN POPULAR, WELL-KNOWN TRADING STRATEGIES MAY BE INCORPORATED INTO PERSONAL TRADING SYSTEMS. OMEGA RESEARCH, INC. IS NOT ENGAGED IN RENDERING ANY INVESTMENT OR OTHER PROFESSIONAL ADVICE. IF INVESTMENT OR OTHER PROFESSIONAL ADVICE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL SHOULD BE SOUGHT.

Contents

INTRODUCTION

Welcome to Volume 4

Welcome to Volume 4 of the *Omega Research System Trading and Development Club*, the revolutionary learning tool offered by Omega Research to help you make the most of your trading potential.

This fourth volume of the *Omega Research System Trading and Development Club* contains 10 new systems we've created to help you get started developing your own systems.

In this volume, Appendix A, "Volume in Review," discusses the results of the recent survey you completed. We were very pleased by the overwhelming response we received to the survey and you will start to see the changes based on your suggestions in Volume 5. Thank you for your feedback!

The systems in Volume 4 incorporate popular concepts such as exponential moving averages, momentum retracement, price distribution, and more. We identify the most common problems with these types of systems and provide techniques to overcome them as well as present different ways of using them. By reviewing and testing these 10 systems, you'll be able to understand these techniques and use them, or a variation of them, in your own systems.

This book walks you step-by-step through the process we went through to develop these systems, from coming up with a feasible trading idea, to writing down your trading rules, to writing them in EasyLanguage, to taking into consideration money management and risk control factors. Our goal is for you to understand why we wrote these systems the way we did.

We recommend you study the EasyLanguage techniques we used, look at the System Report, read about the factors we took into account to deem that a system may have merit, and think about how you can combine some of these popular ideas with your own to develop a system *you* want to trade.

***IMPORTANT NOTICE*: The trading systems in this book are examples only, and have been included solely for educational purposes. Omega Research does not recommend that you use any such trading system, as the use of any such trading system does not guarantee that you will make profits, increase profits, or minimize losses. The sole intended uses of the trading systems included in this book are to demonstrate the ways in which EasyLanguage can be used to design personal trading systems and to show some examples of how certain popular, well-known trading strategies may be incorporated into personal trading systems.**

Contents at a Glance

This book begins by discussing some of the broader concepts of system development before moving on to the description of each system. We grouped the different types of systems together. The contents are:

IMPORTANT NOTE: *We suggest you read the book from front to back because there is instructional material in each section and it is not repeated throughout.*

Additional Educational Services

Omega Research is committed to enhancing individual trading potential through quality education. To learn more about system trading, an Omega Research product, or EasyLanguage, visit our web site at **www.omegaresearch.com** or call **(800) 439-7995** (outside US 305-551-9991) and ask about the following educational services:

Workshops

Omega Research offers a variety of workshops on the products and technical analysis. Workshops are an excellent way to learn how to use the products, learn about technical analysis and system trading and/or EasyLanguage. Spend a day with a Product Training Specialist and exchange ideas with other users like yourself. All workshops provide a 100% satisfaction guarantee. Call now for more information or to register—space is limited!

EasyLanguage Resource Center

One of the best ways to learn is by example, and the EasyLanguage Resource Center on our web site is an excellent source of examples. In this Resource Center, we list all the analysis techniques—indicators and trading systems—published in the *Technical Analysis of Stocks and Commodities* magazine, as well as popular analysis techniques worth taking a look at. Access to this Resource Center is free of charge. Feel free to download and review any of the analysis techniques and their descriptions. Our web site address is **www.omegaresearch.com**.

Getting Started

To begin reviewing your systems, transfer the analysis techniques into your TradeStation library and then apply the system you want to review to a chart. Use the System Report to view the system results and take a look at the EasyLanguage instructions by opening the system in the PowerEditor.

To transfer the analysis techniques into TradeStation:

1. Place the System Trading and Development Club CD in the CD-ROM drive.
2. Start the PowerEditor. In Windows 95, click **Start**, choose **Programs**, choose **Omega Research** and choose **TradeStation PowerEditor**. In Windows 3.x, choose **TradeStation PowerEditor** from the Omega Research program group.
3. In the PowerEditor, use the **File - Open** menu sequence.
4. Click **Transfer**.
5. Select the **Transfer analysis techniques FROM EasyLanguage Archive File** option and click **OK**.
6. Click **Scan**.
7. In the **Enter drive letter to scan** edit box, enter the drive letter for your CD-ROM drive (normally D), and click **OK**. The ELA file on the CD is placed in the list.
8. Choose **JULAUG98.ELA** from the list and click **OK**.
9. In the **Transfer** dialog box, select **Transfer All** and click **OK**.
10. Once the files are transferred and verified, a dialog box appears informing you that the transfer was performed successfully. Click **OK**.

For your convenience, the names of the systems in this volume all begin with STAD4. You can now open the systems in the PowerEditor and view the EasyLanguage instructions and/or apply them to a chart in TradeStation. You can remove your CD from the CD-ROM drive and store it in a safe place. As you apply the systems and work with them, refer to this book for detailed explanations of the systems and the EasyLanguage used to create them. For instructions on applying systems and viewing the System Report, please refer to your *TradeStation User's Manual*.

Note to SuperCharts 4 Users: *To transfer the systems into SuperCharts, use the* ***Tools - QuickEditor*** *menu sequence and select* ***Transfer****. Keep in mind, however, that although you can apply the systems in SuperCharts, you will not be able to view the EasyLanguage instructions in the QuickEditor. This is because the systems were designed in the PowerEditor. Also, if you are using SuperCharts End of Day, some of the systems will not apply as they are designed for intraday trading. Since the purpose of the Club is to provide you with a learning tool, and viewing the EasyLanguage instructions is an essential part of this learning process, the use of this club for SuperCharts users is limited.*

Note to TradeStation or SuperCharts 3.x Users: *The systems for the Club were designed using TradeStation 4. As such, some of the features used, such as automatic drawing of trendlines and/or text, are not available in previous versions of TradeStation (or SuperCharts). An effort is made to provide a variety of systems that incorporate both long standing and new features; however, keep in mind that as new features are developed, we will naturally want to showcase and educate users on these features; therefore, users of the most recent version of our software will be able to make the most use of the Club.*

Obtaining Technical Support

Depending on your question, there are two resources at your disposal: the EasyLanguage Support Department and the STAD Club E-Mail Address.

EasyLanguage Support Department

The EasyLanguage Support Department provides EasyLanguage support via fax and is designed to help you troubleshoot an analysis technique or trading system you are currently working on. For example, if you are incorporating a trading system from the Club into your own and have a question about the implementation, the EasyLanguage Support Department can answer it.

Please keep in mind that while this department can answer any EasyLanguage question, it cannot answer questions about the STAD Club specifically, such as the theory behind a system in the Club, why a system was developed a certain way, or why the system is not performing as you expect it to, etc.

Fax Number:

(305) 221-6831

E-Mail Address:

easylang@omegaresearch.com

Be sure to include the following information in your fax:

- Name
- Security Block or Customer ID Number
- Telephone Number
- Fax Number
- Product you own
- EasyLanguage instructions you are working on
- Detailed description of your problem

Please allow 48 hours for a response.

STAD Club E-Mail Address

Another resource at your disposal is the STAD Club e-mail address.

Please realize that when you send a message to this e-mail address, you will not receive a response directly; your message will be reviewed and the answer incorporated into the next volume of the STAD Club, when applicable. Therefore, if you need technical support on EasyLanguage, please use the above fax number or e-mail address.

stadclub@omegaresearch.com

Please send any comment, suggestion, or question regarding the systems in the Club to the STAD Club e-mail address, and each subsequent volume we will publish the most common suggestions and questions.

CHAPTER 1

General System Development Concepts

Optimization and its value is a source of continuing debate. Those against optimization argue that the process is really fitting the system to the sample data and therefore a useless process that simply gives you the best system for the specific sample of data.

The other side of the argument; however, contends that while there is a risk in the process of optimization, the process itself is valuable when performed properly and with its purpose and limitations kept clearly in mind.

We believe that optimization, used correctly, is an integral part of system development; however, the risks of over-optimization are real and care must be taken to maximize the usefulness of optimization while minimizing the risks.

This chapter discusses optimization, highlights the risks and signs of over-optimization, and should help you use this technique effectively.

In This Chapter

What Is Optimization?

Optimization is the process of testing different inputs and stops to fine-tune a trading system. You use historical data to test the effects of slight changes in your system's criteria.

When you optimize a system, TradeStation runs a series of tests based on different values for system inputs and/or stops and then automatically picks the parameters that yielded the best results according to the criteria you specify.

When you optimize a system, you are not limited to optimizing based on the best net profit. You can optimize for return on account, overall drawdown, or on any of the fields in the System Report, as discussed in the section titled, "Choosing the Criteria for Best Result."

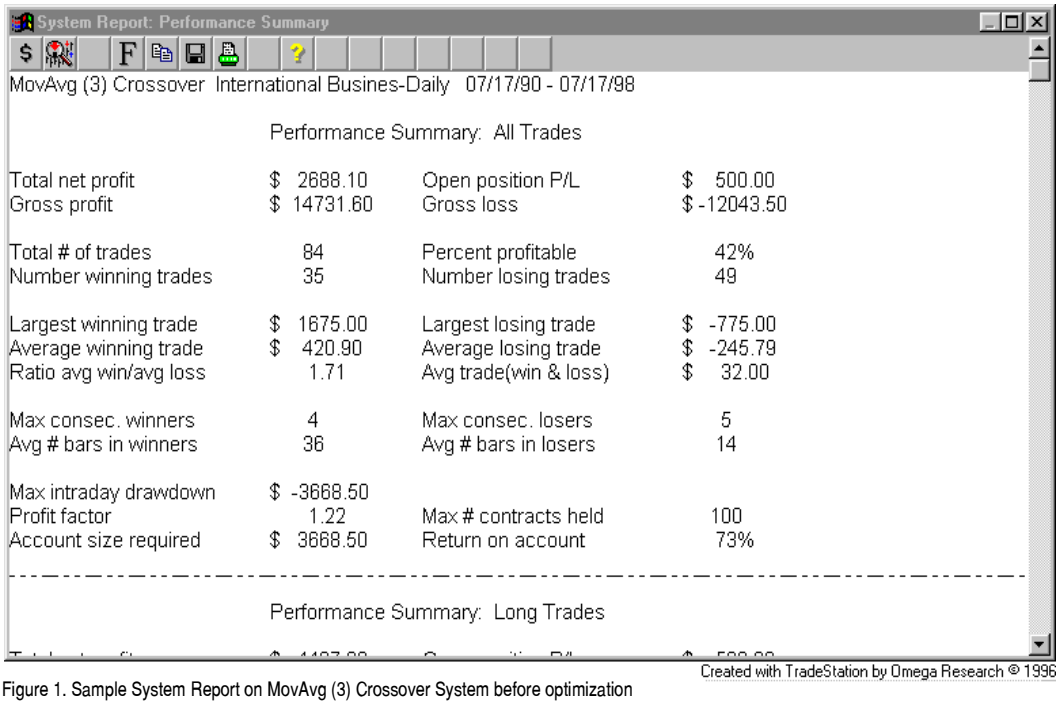
What Can Be Optimized?

As described next, the two elements of a trading system that can be optimized are *simple numeric inputs* and *stops*.

Simple Numeric Inputs

For examples of simple numeric inputs, take a look at the MovAvg (3) Crossover system, one of TradeStation's built-in systems. This system uses the inputs Length1, Length2, and Length3. The input Length1 refers to the number of bars used to calculate the fastest moving average, Length2 the second fastest moving average, and Length3 the third fastest moving average.

TradeStation's default values for this system are 4, 9, and 18, respectively, which when we tested it on a daily IBM chart generated a net profit of $2,688.10. Figure 1 shows the resulting System Report:


System Report: Performance Summary

MovAvg (3) Crossover International Busines-Daily 07/17/90 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 2688.10	Open position P/L	$ 500.00
Gross profit	$ 14731.60	Gross loss	$ -12043.50
Total # of trades	84	Percent profitable	42%
Number winning trades	35	Number losing trades	49
Largest winning trade	$ 1675.00	Largest losing trade	$ -775.00
Average winning trade	$ 420.90	Average losing trade	$ -245.79
Ratio avg win/avg loss	1.71	Avg trade(win & loss)	$ 32.00
Max consec. winners	4	Max consec. losers	5
Avg # bars in winners	36	Avg # bars in losers	14
Max intraday drawdown	$ -3668.50		
Profit factor	1.22	Max # contracts held	100
Account size required	$ 3668.50	Return on account	73%

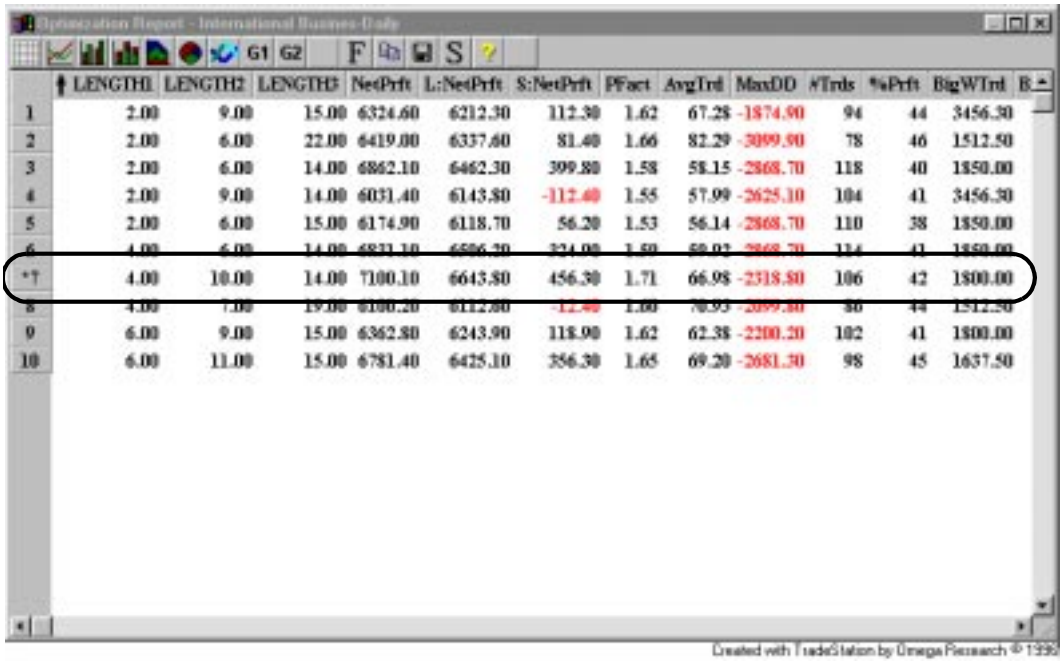
Performance Summary: Long Trades

Created with TradeStation by Omega Research © 1996

Figure 1. Sample System Report on MovAvg (3) Crossover System before optimization

However, you may want to test various combinations of input values for the MovAvg (3) Crossover System. For example, for Length1 you could test the values from 2 to 6, for Length2 the values from 6 to 12, and for Length3 the values from 14 to 22, all in increments of one.

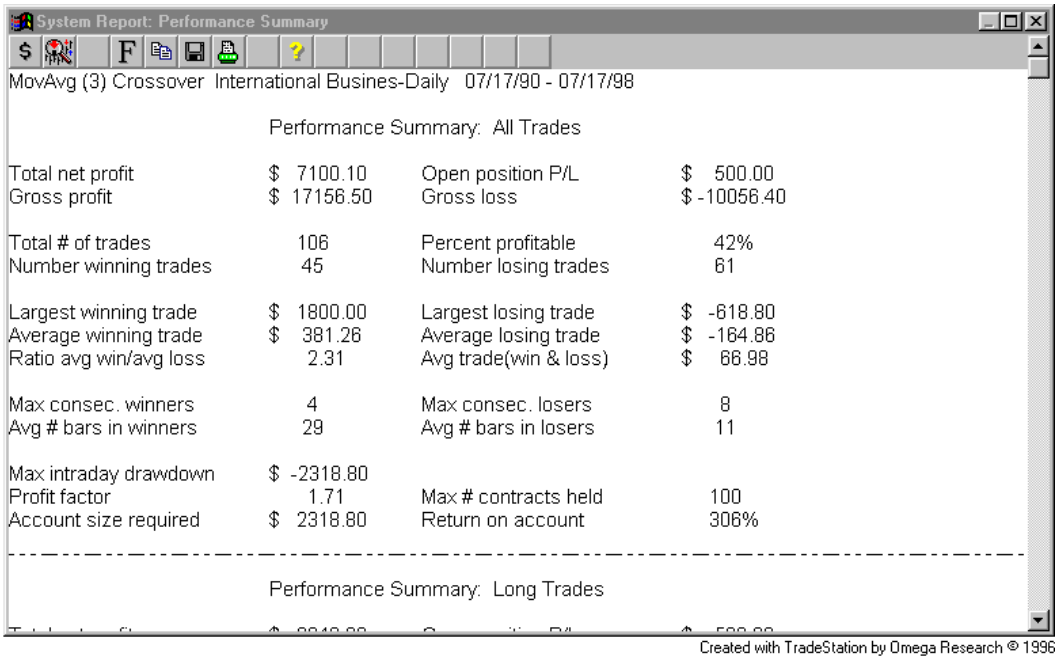
Figure 2 shows a sample Optimization Report for such a test. For this test, we optimized for net profit. In other words, we wanted to find which lengths result in the highest net profit. In this case, the lengths 4, 10 and 14 were the most profitable, as is indicated by the asterisk next to the test number. Notice the net profit rose to $7,100.10.



	LENGTH1	LENGTH2	LENGTH3	NetPrft	L:NetPrft	S:NetPrft	PFact	AvgTrd	MaxDD	#Trds	%Prft	BigWTrd
1	2.00	9.00	15.00	6324.60	6212.30	112.30	1.62	67.28	-1874.90	94	44	3456.30
2	2.00	6.00	22.00	6419.00	6337.60	81.40	1.66	82.29	-3099.90	78	46	1512.50
3	2.00	6.00	14.00	6862.10	6462.30	399.80	1.58	58.15	-2868.70	118	40	1850.00
4	2.00	9.00	14.00	6031.40	6143.80	-112.40	1.55	57.99	-2625.10	104	41	3456.30
5	2.00	6.00	15.00	6174.90	6118.70	56.20	1.53	56.14	-2868.70	110	38	1850.00
6	4.00	6.00	14.00	6831.10	6506.30	324.80	1.69	69.93	-2868.70	114	42	1850.00
*7	4.00	10.00	14.00	7100.10	6643.80	456.30	1.71	66.98	-2318.80	106	42	1800.00
8	4.00	7.00	19.00	6100.20	6112.80	-12.60	1.60	70.93	-2099.80	86	44	1512.50
9	6.00	9.00	15.00	6362.80	6243.90	118.90	1.62	62.38	-2200.20	102	41	1800.00
10	6.00	11.00	15.00	6781.40	6425.10	356.30	1.65	69.20	-2681.30	98	45	1637.50

Figure 2. Optimization Report for the MovAvg (3) Crossover System; the three moving average lengths were optimized

Figure 3 shows the improved System Report generated by the optimized system.



System Report: Performance Summary

MovAvg (3) Crossover International Busines-Daily 07/17/90 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 7100.10	Open position P/L	$ 500.00
Gross profit	$ 17156.50	Gross loss	$ -10056.40
Total # of trades	106	Percent profitable	42%
Number winning trades	45	Number losing trades	61
Largest winning trade	$ 1800.00	Largest losing trade	$ -618.80
Average winning trade	$ 381.26	Average losing trade	$ -164.86
Ratio avg win/avg loss	2.31	Avg trade(win & loss)	$ 66.98
Max consec. winners	4	Max consec. losers	8
Avg # bars in winners	29	Avg # bars in losers	11
Max intraday drawdown	$ -2318.80		
Profit factor	1.71	Max # contracts held	100
Account size required	$ 2318.80	Return on account	306%

Performance Summary: Long Trades

Figure 3. Resulting System Report generated by optimized system

Stops

Stops enable you to set a specific dollar amount or percentage of open profits that you are willing to risk before a trade is automatically closed out. The types of stops that can be optimized in TradeStation include money management stops, $ risk trailing stops, breakeven stops, and % risk trailing stops.

Although profit targets are not a type of stop, they are included in the same dialog as stops because profit targets and stops can be optimized in exactly the same way.

Determining the Number of Tests

The range of values you specify for inputs and stops when you optimize a trading system determines the number of tests TradeStation will perform. The more values for inputs or stops that are optimized, the more tests must be run.

For example, if you want to optimize only one input for the system you are testing, and six different values are to be used for that input, TradeStation will perform six tests (1 x 6 = 6). If you optimize two inputs, and you decide to test six different values for each input, the number of tests increases exponentially to 36 (6 x 6 = 36). Each possible combination is tested to determine which performs best. If, in addition to the two inputs, you also included a money management stop with six different values, the number of tests would climb to 216 (6 x 6 x 6).

Depending on the number of tests you run, an optimization can take seconds, minutes, or even hours. You can cut down the time required for testing by either reducing the number of values being tested for each input or stop, or reducing the number of inputs or stops being optimized. For example, if you are optimizing a system with three unrelated entry signals, you should optimize the inputs for each entry signal separately. If the inputs are related (so that it's not logical to optimize them separately), you can increase the size of the increment to reduce the number of tests. For example, you might decide to test the input values 10, 20, 30, and 40 rather than *all* the values between 10 and 40. Then, when you determine the range of input values that performs best, you can re-optimize using this reduced range and a smaller increment.

Choosing the Criteria for Best Result

TradeStation enables you to choose the criterion for the best result when you optimize a trading system. The default criterion for the best result of a system optimization is the value or combination of values that produces the highest total net profit. However, the value that produces the highest net profit is not always the best result. Would the value that yielded the highest net profit but also produced the largest drawdown and the smallest winning percentage necessarily be the best choice?

TradeStation offers more than 60 criteria for you to choose from for determining the best result of an optimization. For example, instead of total net profit, you could select Average trade (Wins + Losses), Percent profitable, or Profit factor (how many dollars won for each dollar lost). The criteria for best result are calculated for three categories: All, Long, and Short. You can, for example, optimize for Total net profit in long positions only, Ratio of average win to average loss in short positions only, or Profit factor in both long and short positions. (The criteria within each category are identical.)

Avoiding the Over-Optimization Trap

If you over-optimize your trading system's inputs or stops, you run the risk of curve-fitting the parameters to the data you used to test your system. Since it is highly unlikely that a lengthy series of price fluctuations will repeat itself exactly in the future, you must avoid the trap of over-optimization. Following are some suggestions for constructive optimization as opposed to curve-fitting optimization:

1. Do not rely on optimization to create a trading system. Optimization should be one of the last steps you take when you are developing a system. It is not time to optimize a trading system until the system is already profitable as it stands.
2. Develop a trading system that is based on a logical trading idea. Avoid trading systems that yield good optimized results but that are not based on a sound market theory.
3. Keep your trading systems as simple as possible. The markets don't pay you any extra for designing and trading a more complex system.
4. Test your trading idea on several stocks and/or commodities. If your test results are poor on other markets, the system is probably curve-fitted to one set of data.
5. The best value discovered by optimization should have a range of profitable values around it. For example, if a system's optimum moving average value is 20, values on either side of it (e.g. 18, 19, 21, and 22) should also produce good results.
6. Include both backward and forward testing in your system evaluation. The optimal parameters you discovered when testing one set of data should also be profitable on both earlier and later data.

If you optimize your trading systems with these suggestions in mind, you'll be well on your way to effective, meaningful optimization, and you'll avoid ineffective, curve-fitted optimization.

CHAPTER 2

Trending Systems

Trending systems are systems designed for trending markets — they have the following characteristics:

- The systems are designed never to miss the big move; they will either always be in the market or contain stop orders that will stop you into the market.
- They attempt to limit losses during the market's sideways mode; no system will make money in every market condition, but a good system will limit losses in market conditions for which it was not designed.
- Profits are concentrated in a few big trades; they have a low percentage of profitable trades. This makes them psychologically difficult to trade and underscores the importance of never missing a big move.

Even though trending systems are difficult to trade, they are popular — it's human nature to want to cash in on the big moves. In this chapter, we present seven trending systems that differ in their approach but that are all designed to capture big moves and limit losses during directionless and/or volatile phases.

In This Chapter

Directional Movement Index System

The Directional Movement Index (DMI) was created by Welles Wilder and presented in his 1978 book *New Concepts in Technical Trading Systems*. The DMI indicator measures the amount of directional movement or "trendiness" in a market, and is composed of three lines: DMI+ (DMIPlus), DMI- (DMIMinus), and ADX (Average Directional Index). When DMI+ is greater than DMI-, the trend is up; when DMI- is greater than DMI+, the trend is down. ADX is derived from the spread between DMI+ and DMI-. When ADX is rising, the market is considered to be in a trending mode. When ADX is falling, the market is considered to be in a nontrending mode.

The idea behind the Directional Movement Index System is that markets tend to cycle from a nontrending period to a trending period, back to a nontrending period, and so on. The longer a market trades in a nontrending mode, the more likely it is that the ensuing trend will offer a good trading opportunity. We will watch for stocks that have been in a nontrending mode and enter new positions when the stocks begin to trend.

To determine whether or not a stock is currently trending, we'll use the Directional Movement Index (DMI). If ADX is below both the DMI+ and the DMI- lines, then we'll consider the stock as not trending. Once we've determined that the stock is not trending, we'll look at the behavior of the DMI for our entry conditions.

As discussed, the market is not trending when ADX is below both the DMI+ and DMI- lines; however, the market's direction is considered to be up when DMI+ is above DMI- and down when DMI- is above DMI+.Therefore, when DMI+ is greater than DMI-, and ADX crosses above DMI-, we have a buy setup. When DMI- is greater than DMI+, and ADX crosses above DMI+, we have a setup to sell short.

Once a buy setup occurs, we establish our entry point as the high of the setup bar plus one point. When a sell setup occurs, our entry point is set at the low of the setup bar minus one point. The buy setup remains in effect until the entry point is reached or until ADX declines below DMI-; likewise, the sell setup remains in effect until the entry point is reached or until ADX declines below DMI+. Figure 1 shows the Directional Movement Index System applied to a chart along with the DMI Indicator (built into TradeStation).

Note: When reproducing this chart, make sure you remember to match up the inputs in your system and the indicator.

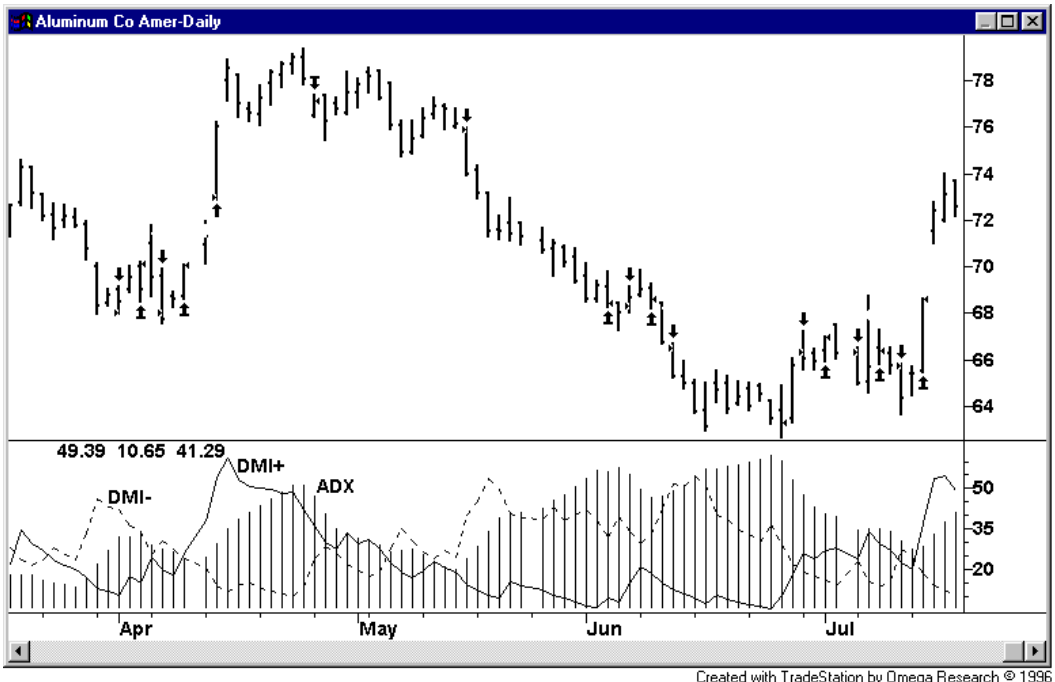


Figure 1. DMI System applied to a daily chart of AA.

When prices rally to our entry point for a buy, we'll set our initial protective stop at the low of the setup bar minus one point. When prices decline to our entry point to sell short, we'll set our initial protective stop at the high of the setup bar plus one point.

We have chosen two ways to exit our position in addition to the initial protective stop. We'll exit our position with whichever exit criteria is met first. First, we will exit either a long or a short position on the next open when ADX reaches 30 and then ticks down (by "ticks down" we mean that ADX is less than it was on the previous bar). Alternatively, we will exit a long position when DMI+ crosses below DMI- or a short position when DMI- crosses below DMI+.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some setup work to calculate the DMI. The setup, entries and exits are described next:

Setup

a) Calculate the DMI values: DMI+, DMI- and ADX.

Long Entries

a) When DMI+ is greater than DMI-, and ADX crosses above DMI-, and we are not in a long position, we have a buy setup.

b) Once a buy setup occurs, we establish our entry point as the high of the setup bar plus one point.

c) The buy setup remains in effect until the entry point is reached or until ADX declines below DMI-.

Short Entries

a) When DMI- is greater than DMI+, and ADX crosses above DMI+, and we are not in a short position, we have a setup to sell short.

b) When a sell setup occurs, our entry point is set at the low of the setup bar minus one point.

c) The sell setup remains in effect until the entry point is reached or until ADX declines below DMI+.

Exit Orders

a) When prices rally to our entry point for a buy, we'll set our initial protective stop at the low of the setup bar minus one point.

b) When prices decline to our entry point to sell short, we'll set our initial protective stop at the high of the setup bar plus one point.

c) We will exit either a long or a short position at the close once we've been in the position at least one bar and when ADX reaches 30 and then ticks down (by "ticks down" we mean that ADX is less than it was on the previous bar).

d) We will exit a long position when we've been in the position at least one bar and when DMI+ crosses below DMI- or a short position when we've been in the position at least one bar and when DMI- crosses below DMI+.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Directional Movement Index (STAD 4: DMI-ADX)

```
Input: DMILen(7), ADXLen(7), ADXExit(30), Trade(10000), Lot(100);
Vars: BuySetup(False), SellSetup(False), BuyPrice(0), SellPrice(0), DPlus(0),
        DMin(0), ADXVal(0), ExitSPrc(0), ExitLPrc(0);

{Calculate DMI and ADX Values}
DPlus = DMIPlus(DMILen);
DMin = DMIMinus(DMILen);
ADXVal = ADX(ADXLen);

{Check for Buy Setup}
If DPlus > DMin AND ADXVal Crosses Over DMin and MarketPosition <> 1 then Begin
        BuyPrice = High + 1 Point;
        BuySetup = True;
        ExitLPrc = Low - 1 Point;
End;

If ADXVal < DMin OR MarketPosition = 1 then
        BuySetup = False;

{Check for Sell Setup}
If DPlus < DMin AND ADXVal Crosses Over DPlus and MarketPosition <> -1 then Begin
        SellPrice = Low - 1 Point;
        SellSetup = True;
        ExitSPrc = High + 1 Point;
End;

If ADXVal < DPlus or MarketPosition = -1 then
        SellSetup = False;

{Place Long Entries}
If BuySetup then
        Buy NumUnits(Trade,Lot) Shares next bar at BuyPrice Stop;

{Place Short Entries}
If SellSetup then
        Sell NumUnits(Trade, Lot) Shares next bar at SellPrice Stop;

{Exit Criteria}
If MarketPosition = 1 then Begin
        BuySetup = False;
        ExitLong at ExitLPrc Stop;
End;

If MarketPosition = -1 then Begin
        SellSetup = True;
        ExitShort next bar at ExitSPrc Stop;
End;

If ( ADXVal[1] >= ADXExit AND ADXVal < ADXVal[1] ) then Begin
        If MarketPosition = 1 and BarsSinceEntry > 1 then
                ExitLong this bar at Close;
        If MarketPosition = -1 and BarsSinceEntry > 1 then
                ExitShort this bar at Close;
End;

If DPlus < DMin and BarsSinceEntry > 1  then ExitLong this bar at Close;
If DPlus > Dmin and BarsSinceEntry > 1 then ExitShort this bar at Close;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default	Description
DMILen	7	The number of bars used in the calculation of the DMI.
ADXLen	7	The number of bars used in the calculation of the ADX.
ADXExit	30	The level to which the ADX has to reach in order for one of the exits to take effect (the ADX reaches 30 and then ticks down).
Trade	10000	The dollar amount of each trade.
Lot	100	The minimum lot size of each transaction.

In addition to these inputs, we define the following variables:

```
Vars: BuySetup(False), SellSetup(False), BuyPrice(0), SellPrice(0), DPlus(0),
      DMin(0), ADXVal(0), ExitSPrc(0), ExitLPrc(0);
```

Setup

Next, we calculate the DMI values and store the results to our variables:

```
DPlus = DMIPlus(DMILen);
DMin = DMIMinus(DMILen);
ADXVal = ADX(ADXLen);
```

Long & Short Entries

If the DMI+ is greater than the DMI- and the ADX crosses over the DMI - and we are not in a long position already, we will set our entry price for a long position to the high of the current bar plus 1 point (stored in the variable BuyPrice). In addition, we set our exit price to the low minus 1 point as well as the variable BuySetup to True, meaning that the buy setup conditions have been met:

Note: 1 point is equivalent to one minimum move of the symbol traded. TradeStation reads the Symbol Universe to determine the value of the minimum move.

```
If DPlus > DMin AND ADXVal Crosses Over DMin and MarketPosition <> 1 then Begin
      BuyPrice = High + 1 Point;
      BuySetup = True;
      ExitLPrc = Low - 1 Point;
End;
```

If the ADX declines back under the DMI- or we enter a long position, we set the BuySetup variable to False. This means we will no longer be looking for entry points:

```
If ADXVal < DMin OR MarketPosition = 1 then
      BuySetup = False;
```

If the DMI- is lower than the DMI- and the ADX crosses over the DMI- and we are not in a short position already, we will set our entry price for a short position to the low of the current bar minus 1 point (stored in the variable SellPrice). In addition, we set our exit price to the high plus 1 point as well as the variable SellSetup to True, meaning that the sell setup conditions have been met:

```
If DPlus < DMin AND ADXVal Crosses Over DPlus and MarketPosition <> -1 then Begin
      SellPrice = Low - 1 Point;
      SellSetup = True;
      ExitSPrc = High + 1 Point;
End;
```

We will set the SellSetup variable to False if the ADX declines back under the DMI- or we enter a short position. This means we will no longer be looking for short entry points.

```
If ADXVal < DPlus or MarketPosition = -1 then
        SellSetup = False;
```

If the variable BuySetup is true, then we will place an order to buy on the next bar when the market reaches the price specified by the BuyPrice variable. In order to determine the number of shares/contracts to trade, we use the function NumUnits. This function calculates the number of units you can trade based on a given dollar number investment and the minimum lot size for the market. For example, if you want to invest $15,500, the market trades in lots of 100 shares, and the price of the stock is $65 a share, NumUnits(15500,100) will return 200. This function is described in detail at the end of this chapter, in the section titled, "Investing a Fixed Dollar Amount."

```
If BuySetup then
        Buy NumUnits(Trade,Lot) Shares next bar at BuyPrice Stop;
```

The same is done for the short side:

```
If SellSetup then
        Sell NumUnits(Trade,Lot) Shares next bar at SellPrice Stop;
```

Once we enter a long position, we will set the variable BuySetup to False, this to avoid placing additional entry orders. Also, we will place a stop loss exit at the ExitLPrc, this will have the value of the low minus 1 point of the setup bar, as defined previously.

```
If MarketPosition = 1 then Begin
        BuySetup = False;
        ExitLong at ExitLPrc Stop;
End;
```

Once we enter a short position, we will set the variable SellSetup to False, this to avoid placing additional entry orders. Also, we will place a stop loss exit at the ExitSPrc, this will have the value of the high plus 1 point of the setup bar as defined previously.

```
If MarketPosition = -1 then Begin
        SellSetup = True;
        ExitShort next bar at ExitSPrc Stop;
End;
```

Long & Short Exits

If the ADX reaches the value specified in the ADXExit input (30 by default), and then declines once we have been in the position for more than 1 bar, we will exit either our long or short positions:

```
If ( ADXVal[1] >= ADXExit AND ADXVal < ADXVal[1] ) then Begin
        If MarketPosition = 1 and barssinceentry > 1 then
                ExitLong this bar at Close;
        If MarketPosition = -1 and barssinceentry > 1 then
                ExitShort this bar at Close;
End;
```

If the DMI+ declines and crosses under the DMI- we consider the market position to have changed so we will exit our long position (provided we've been in the position at least one bar):

```
If DPlus < DMin and BarsSinceEntry > 1 then ExitLong this bar at Close;
```

If DMI Minus crosses under DMI-, we will exit our short position (provided we've been in the position at least one bar):

```
If DPlus > Dmin and BarsSinceEntry > 1 thenExitShort this bar at Close;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We did not specify a default number of shares to trade per order. Instead, we want to focus on what we believe to be a much improved method for determining the number of shares to trade. Please see the section at the end of this chapter, titled, “Investing a Fixed Dollar Amount,” for an explanation of how we are calculating the number of shares to trade.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

We also did not enable a money management stop or a trailing stop for this system. Our initial protective stop for a long position is set at the low of the setup bar plus one point; our initial stop for a short position is set at the high of the setup bar plus one point.

Rather than enabling a trailing stop, we decided to demonstrate a system that exits on signals generated by indicators. We’ll exit our positions with one of two possible strategies. When ADX reaches 30 and ticks down, we’ll exit on the next open. Alternatively, we’ll exit a long position when DMI+ crosses below DMI- and exit a short position when DMI- crosses below DMI+ (we’ll exit with whichever strategy occurs first).

In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored. This is also the case when we’re in a short position, and market conditions generate another short entry order.

Testing & Improving

Figure 2 shows a sample System Report for the DMI-ADX trading system.

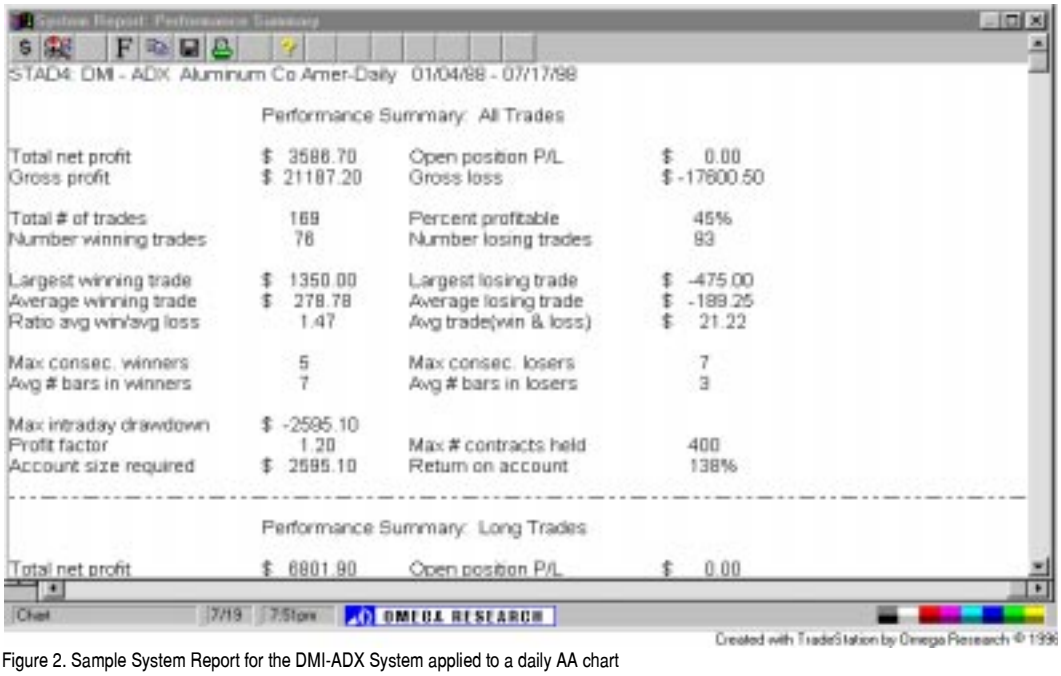
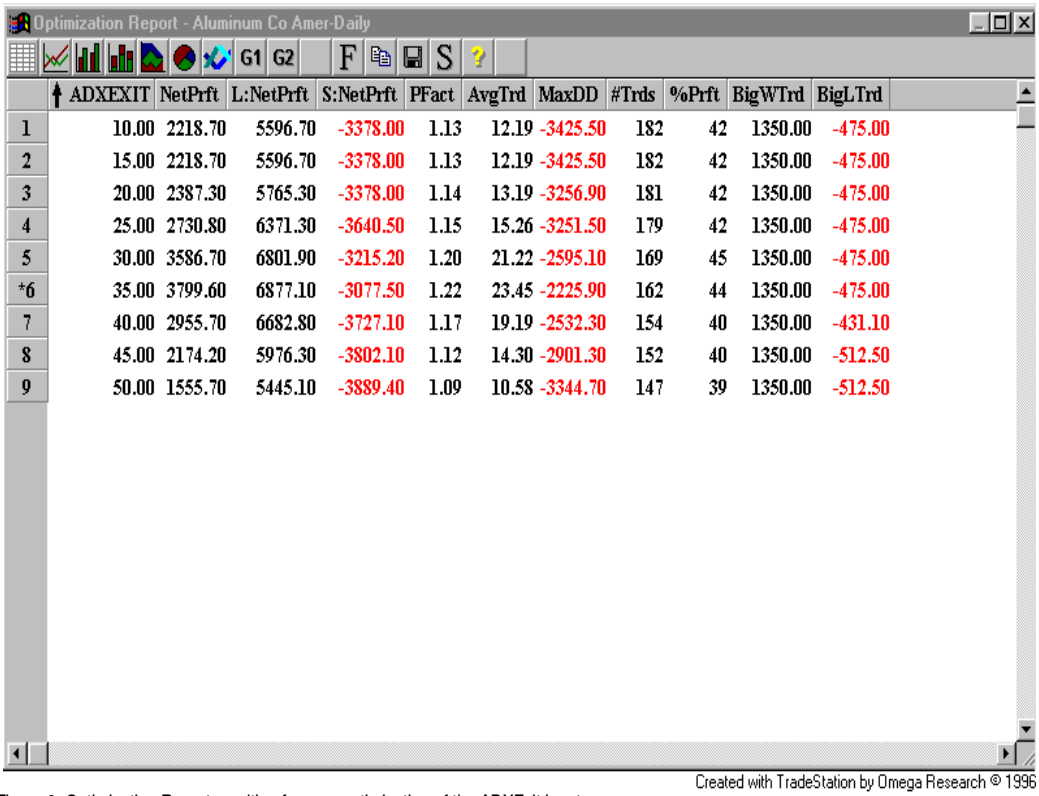


Figure 2. Sample System Report for the DMI-ADX System applied to a daily AA chart

The total net profit of only $3,586.70 is misleading because the system actually made $6,801.90 on long trades and lost $3,215.20 on short trades. The system was tested from January of 1988 to July of 1998, a period in which stocks were in an unprecedented bull market. It’s not surprising that this system (a trend-following system) performed poorly when it sold short.

Since very few of us actually sell stock short (generally using sell signals to exit long positions), we're not overly concerned about the system's poor performance on the short side. A result of 45% profitable is not bad for a trend-following system. The average win was 1.47 times as big as the average loss.

The only parameter we optimized was the level that ADX had to reach before our ADX turndown would be a signal for us to exit our position on the next open. The level 35 generated the largest net profit, profit factor, and average trade, while producing the smallest maximum drawdown. Figure 3 shows the resulting Optimization Report.



	ADXEXIT	NetPrft	L:NetPrft	S:NetPrft	PFact	AvgTrd	MaxDD	#Trds	%Prft	BigWTrd	BigLTrd
1	10.00	2218.70	5596.70	-3378.00	1.13	12.19	-3425.50	182	42	1350.00	-475.00
2	15.00	2218.70	5596.70	-3378.00	1.13	12.19	-3425.50	182	42	1350.00	-475.00
3	20.00	2387.30	5765.30	-3378.00	1.14	13.19	-3256.90	181	42	1350.00	-475.00
4	25.00	2730.80	6371.30	-3640.50	1.15	15.26	-3251.50	179	42	1350.00	-475.00
5	30.00	3586.70	6801.90	-3215.20	1.20	21.22	-2595.10	169	45	1350.00	-475.00
*6	35.00	3799.60	6877.10	-3077.50	1.22	23.45	-2225.90	162	44	1350.00	-475.00
7	40.00	2955.70	6682.80	-3727.10	1.17	19.19	-2532.30	154	40	1350.00	-431.10
8	45.00	2174.20	5976.30	-3802.10	1.12	14.30	-2901.30	152	40	1350.00	-512.50
9	50.00	1555.70	5445.10	-3889.40	1.09	10.58	-3344.70	147	39	1350.00	-512.50

Figure 3. Optimization Report resulting from an optimization of the ADXExit input

Figure 4 shows the bar graph view of the above Optimization Report. It shows profits increasing as the ADX exit level increases from 10 to 35 and decreasing from 40 to 50. Notice that the optimum exit level of 35 also had strong-performing values to both its left and right.

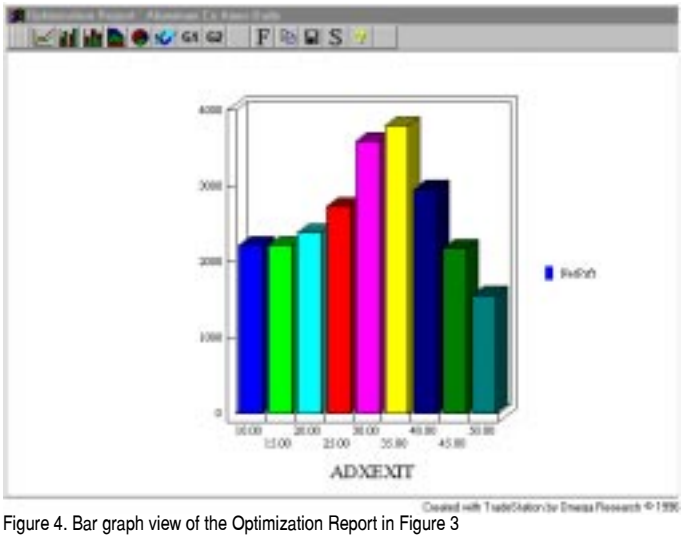


Figure 4. Bar graph view of the Optimization Report in Figure 3

Suggestion for Improvement

When designing the system, we arbitrarily chose values of 7 for the DMI length and the ADX length. Before trading this system with our hard-earned money, we would want to optimize the DMI and ADX values on a large data set and then check to see that the optimized values performed well on our-of-sample data. The out-of-sample data would include earlier and later dates of the same stock (backward and forward testing) as well as testing the optimized parameters on other stocks and perhaps some commodities as well. While we would not expect the optimized parameters to perform as well on the out-of-sample data as they did on the sample, we would require that they be generally profitable in order for us to trade the system.

Keltner Channel System

The construction of a Keltner Channel begins with the calculation of a moving average and an average true range. Then an upper band is plotted at the moving average plus a multiple of the average true range, and a lower band is plotted at the moving average minus a multiple of the average true range. Generally, the length selected for the moving average is also the length selected for the number of bars included for the average true range. The multiple of the average true range that is added to and subtracted from the moving average determines the sensitivity of the Keltner Channel.

Adding 1x the average true range to the moving average to determine the top band of the channel, and subtracting 1x the average true range to determine the bottom of the channel results in a relatively narrow channel. Adding 5x the average true range to the moving average to determine the top band and subtracting 5x the average true range to determine the bottom band produces a very wide channel. For this system, we selected a 10-bar simple moving average of closes, a 10-bar average true range, and a 1.2 multiple of the average true range.

Our Keltner Channel System attempts to profit from trending price moves while limiting “whipsaw” losses (“whipsaw” losses occur when a sensitive entry indicator is triggered repeatedly causing several relatively small losses within a short period of time). Buying when a market closes above a sensitive moving average (for example, a 10-bar average) and selling short when a market closes below the moving average usually results in an unacceptable number of “whipsaw” losses. Our Keltner Channel attempts to mitigate that problem by requiring prices to rally above the top band of the channel for a buy and to decline below the lower band of the channel for a sell.

Of course, the advantage of filtering out many of the “whipsaw” losses comes with a disadvantage. Our buys will be executed at a higher price (above the top band), and our sells will be executed at a lower price (below the bottom band) than they would be if we bought just above or sold just below the moving average. Our goal is to save more money by reducing “whipsaw” losses than we spend by buying higher and selling lower.

Our setup to buy is a close above the top line of the channel, and our setup to sell short is a close below the bottom line of the channel. Figure 5 shows the Keltner Channel System applied to a weekly American Express chart.

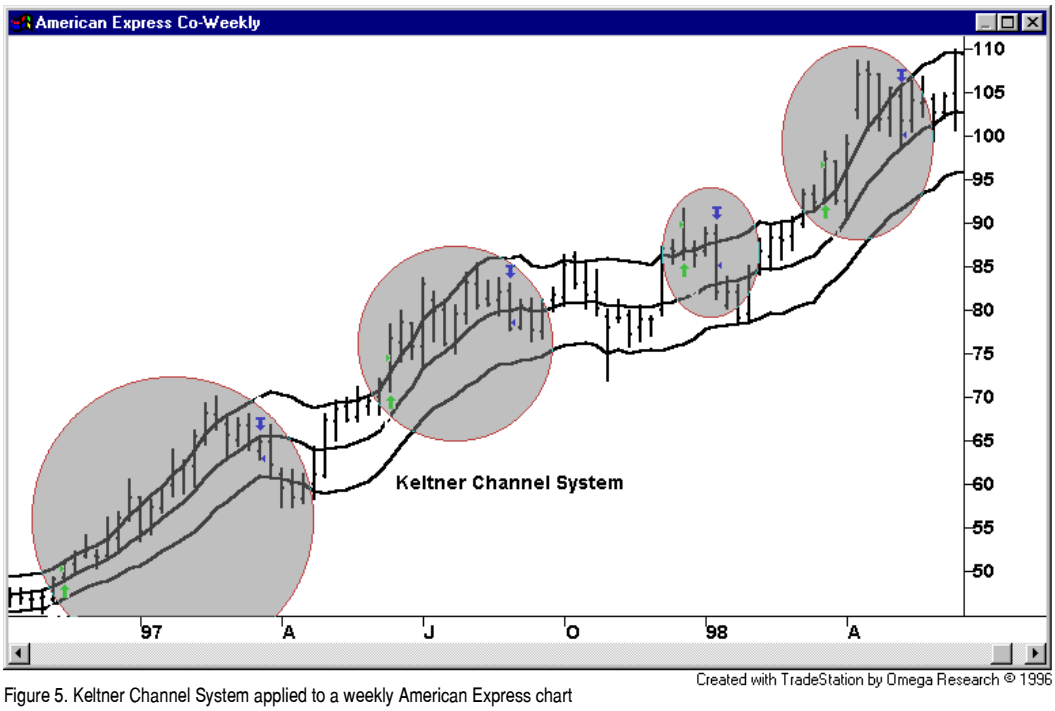


Figure 5. Keltner Channel System applied to a weekly American Express chart

With a buy setup in place, our next step is to determine our entry point. First we measure the distance between the top band of the channel and the moving average. Then we add 50% of that distance to the high of the setup bar for our buy point. When we have a setup to sell short, we first measure the distance between the moving average and the bottom band of the channel. Then we subtract 50% of that distance from the low of the setup bar to determine our sell point.

We’ll keep the entry orders active for 5 bars; if the entry price is not reached within 5 bars of the cross over, the orders are cancelled and we will wait for another setup to occur.

After we enter a position, we’ll set a money management stop to limit our loss to a specified dollar amount if the market moves against us. To manage the trade, we’ll set a trailing stop at the 4-bar extreme. Our exit, if we are not stopped out by our money management stop or our trailing stop, will be on a close below the moving average (in the case of a long position) and on a close above the moving average (for a short position).

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some setup work to calculate the channel. The setup, entries and exits are described next:

Setup

a) Calculate a channel using the 10-bar simple moving average of closing prices and a 10-bar average true range with a 1.2 multiple.

b) Calculate the channel’s range.

Long Entries

a) Check for a close above the top line of the channel.

b) Once a close is above the top line, measure the distance between the top band of the channel and the moving average and add 50% of that distance to the high of the setup bar. This is our buy point, buy when the price reaches this point.

c) Keep the order active for 5 bars.

Short Entries

a) Check for a close below the bottom line of the channel.

b) Once a close is below the bottom line, measure the distance between the moving average and the bottom band of the channel. Then, subtract 50% of that distance from the low of the setup bar. This is our sell point, sell when the price reaches this point.

c) Keep the order active for 5 bars.

Exit Orders

a) Once we enter a position, set a trailing stop at the 4-bar extreme. We'll exit our long position when the close penetrates the lowest low of the last 4 bars and we'll exit our short position when the close penetrates the highest high of the last 4 bars.

b) Also, exit a long position on a close below the moving average and exit a short position on a close above the moving average (for a short position).

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Keltner Channel (STAD 4: Keltner Chan)

```
Inputs: Price(Close), Length(10), Const(1.2), ChanPcnt(.5), Trade(10000), Lot(100);
Vars: KCU(0), KCL(0), ChanRng(0), AvgVal(0), AvgRange(0), SetBar(0), CountL(0), CountS(0);

{Assignments of Keltner calculations}
AvgVal = Average(Price, Length);
AvgRange = Average(TrueRange, Length);
KCU = AvgVal + AvgRange * Const;
KCL = AvgVal - AvgRange * Const;
ChanRng = (KCU - KCL) / 2;

{Accumulates to count the bars after the SetUps below}
CountL = CountL + 1;
CountS = CountS + 1;

{Buy Criteria Evaluation}
If Price Crosses Above KCU then Begin
        SetBar = High;
        CountL = 1;
End;

If Price > KCU AND CountL <= 5 then
        Buy NumUnits(Trade, Lot) Shares Next Bar at SetBar + (ChanRng * ChanPcnt) Stop;
```

```
{Sell Criteria Evaluation}
If Price Crosses Below KCL then Begin
        SetBar = Low;
        CountS = 1;
End;

If Price < KCL AND CountS <= 5 then
        Sell NumUnits(Trade, Lot) Shares Next Bar at SetBar - (ChanRng * ChanPcnt) Stop;

{System Stops}
If Close Crosses Below AvgVal thenExitLong this Bar on Close;
If Close Crosses Above AvgVal thenExitShort this Bar on Close;

{Trailing Stops}
ExitLong Next Bar at Lowest(Low, 4) Stop;
ExitShort Next Bar at Highest(High, 4) Stop;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default	Description
Price	Close	Price used to calculate the moving average (on which the channels will be based).
Length	10	Length, expressed in bars, used to calculate the moving average and average true range (in order to calculate Keltner Channel)
Const	1.2	Multiplier used to specify the width of the channel
ChanPcnt	.5	Percentage of channel range to add to the high to determine the long entry point and to subtract from the low to determine the short entry point
Trade	10,000	Dollar amount willing to invest in each transaction
Lot	100	Minimum lot size for each transaction

In addition to these inputs, we define the following variables:

```
Vars: KCU(0), KCL(0), ChanRng(0), AvgVal(0), AvgRange(0), SetBar(0), CountL(0),
        CountS(0);
```

Setup

We calculate the moving average of the closing price as well as of the true range for the last 10 bars, and store the resulting values in the variables AvgVal and AvgRange, respectively.

```
AvgVal = Average(Price, Length);
AvgRange = Average(TrueRange, Length);
```

Then, we multiply AvgRange by the multiplier and add the resulting value to AvgVal. This gives us the upper band of the Keltner Channel. Likewise, we perform the same multiplication again and this time subtract the resulting value from AvgVal. This gives us the lower band of the Keltner Channel. Finally, we calculate the channel range by subtracting the lower band from the top band and dividing the resulting value by 2:

```
KCU = AvgVal + AvgRange * Const;
KCL = AvgVal - AvgRange * Const;
ChanRng = (KCU - KCL) / 2;
```

We use two counters, CountL and CountS. We increment these with each bar and reset them each time a buy setup or sell setup occurs, respectively. This way, we can keep a count and use them to keep buy and sell orders active for 5 bars:

Counters are a simple and very common mechanism used in EasyLanguage by which you can keep track of how many times an event occurs, or how many bars have been evaluated since an event occurred.

You set a variable to 1 when the first event occurs and then increment the variable as each event occurs or bar is evaluated.

```
CountL = CountL + 1;
CountS = CountS + 1;
```

Long Entries

Whenever the close (represented here by the input Price) crosses above the upper Keltner Channel band, we will save the high of the bar in the variable SetBar and we will reset the variable CountL to 1. This variable will be incremented on every bar thereafter by the above instructions.

```
If Price Crosses Above KCU then Begin
        SetBar = High;
        CountL = 1;
End;
```

If the close is above the upper Keltner Channel band, and it has been five or less bars since the cross over, we will place a buy stop order at the high of the set up bar, which is the bar on which the close crossed over the upper Keltner Channel band (this value is stored in the variable SetBar) plus 50% of the channel range.

```
If Price > KCU AND CountL <= 5 then
        Buy NumUnits(Trade,Lot) Shares Next Bar at SetBar + (ChanRng * ChanPcnt) Stop;
```

Short Entries

Whenever the close crosses under the lower Keltner Channel band, we store the low in the variable SetBar and reset the variable CountS to 1. As with the variable CountL, this variable is incremented on every bar thereafter by the previous instructions, CountS = CountS + 1.

```
If Price Crosses Below KCL then Begin
        SetBar = Low;
        CountS = 1;
End;
```

If the price is less than the lower Keltner Channel band, and it has been five or less bars since the cross under, we will place a sell stop order at the low of the set up bar, which is the bar on which the close crossed under the lower Keltner Channel band (this value is stored in the variable SetBar) minus 50% of the channel range.

```
If Price < KCL AND CountS <= 5 then
        Sell NumUnits(Trade,Lot) Shares Next Bar at SetBar - (ChanRng * ChanPcnt) Stop;
```

Long & Short Exits

We will exit any long position if the close crosses under the moving average of the close. Also, we will cover any short positions if the close crosses over the moving average of the close.

```
If Close Crosses Below AvgVal thenExitLong This Bar on Close;
If Close Crosses Above AvgVal then ExitShort This Bar on Close;
```

Finally, we will place a trailing stop for long positions at the lowest low of the last four bars. Likewise, we will place a trailing stop to cover our short positions at the highest high of the last four bars.

```
ExitLong Next Bar at Lowest(Low, 4) Stop;
ExitShort Next Bar at Highest(High, 4) Stop;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified $10,000 as the trade size and 100 shares as the lot size.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

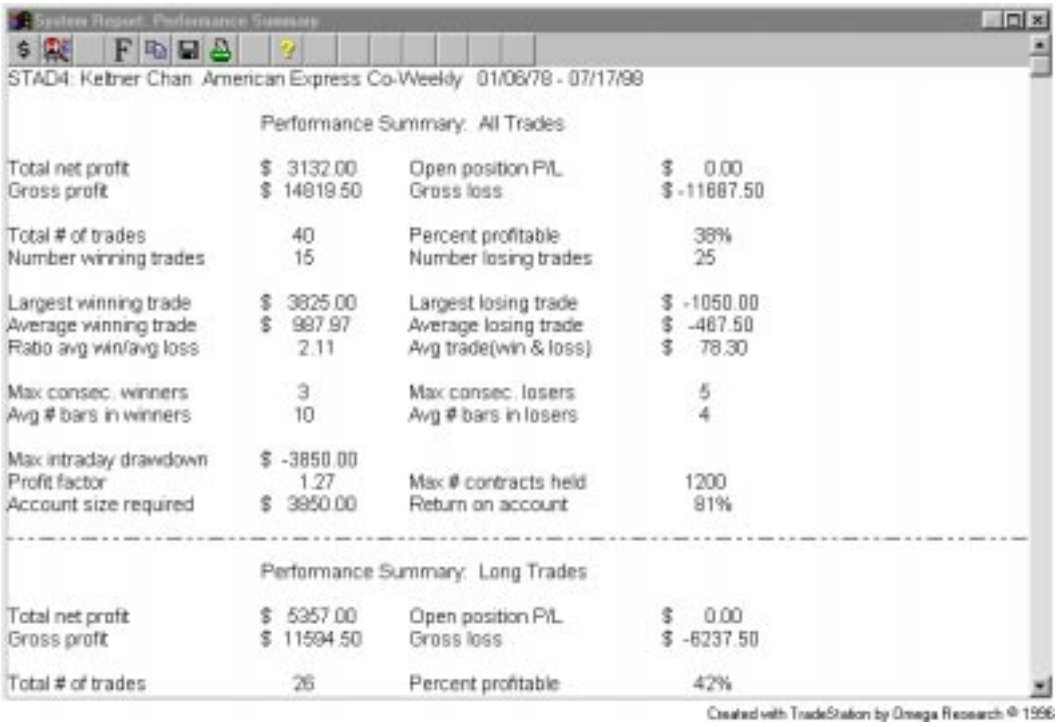
Under the **Stops** tab, we enabled a money management stop. The money management stop option holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We set the value to $7.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the **Do not allow multiple entries in the same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Because we tested this system on weekly data, we lengthened the test period from about 10 years to about 20 years. We still generated a smaller number of trades than were generated on systems we tested on daily bars, of course, but at 36 trades these results are statistically significant. Figure 6 shows the sample System Report.


System Report: Performance Summary

STAD4: Keltner Chan American Express Co-Weekly 01/06/78 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 3132.00	Open position P/L	$ 0.00
Gross profit	$ 14819.50	Gross loss	$ -11687.50
Total # of trades	40	Percent profitable	38%
Number winning trades	15	Number losing trades	25
Largest winning trade	$ 3825.00	Largest losing trade	$ -1050.00
Average winning trade	$ 987.97	Average losing trade	$ -467.50
Ratio avg win/avg loss	2.11	Avg trade(win & loss)	$ 78.30
Max consec. winners	3	Max consec. losers	5
Avg # bars in winners	10	Avg # bars in losers	4
Max intraday drawdown	$ -3850.00		
Profit factor	1.27	Max # contracts held	1200
Account size required	$ 3850.00	Return on account	81%

Performance Summary: Long Trades

Total net profit	$ 5357.00	Open position P/L	$ 0.00
Gross profit	$ 11594.50	Gross loss	$ -6237.50
Total # of trades	26	Percent profitable	42%

Created with TradeStation by Omega Research © 1998

Figure 6. Sample System Report for the Keltner Channel System on a weekly American Express chart

As we have come to expect, the system made money on the long trades and lost money on the short trades during this bull market. With that in mind, 44% profitable for all trades combined, with a ratio of average win to average loss of 1.82 and a profit factor of 1.45 isn't too bad.

Suggestions for Improvement

The most promising way to make a big improvement to the system is to employ a longer-term filter to keep us on the bullish side of this historic uptrend. Granted, we could not know the duration and magnitude of this bull market in advance; however, we would not need to know the extent of the bull market in advance to limit our trading to the long side. Since we tested this system on a weekly chart, we could go just one timeframe higher to a monthly chart, apply a moving average to the monthly data, and only trade in the direction of the monthly moving average.

Of course, we can also fine-tune this system by optimizing the values for the moving average, the average true range, and the multiple of the average true range we use to determine the upper and lower bands.

Other parameters that could improve the system's performance with some attention are the percentage of the distance between the top band of the channel and the moving average that we add to the high of the setup bar to determine our buy point, and the percentage of the distance between the bottom band of the channel and the moving average that we subtract from the low of the setup bar to determine our sell point. We should not assume that the percentages will necessarily be the same for long and short positions.

Linear Regression & Momentum System

The Linear Regression Indicator plots a line through the prices of a stock or commodity in an attempt to minimize the distance between the line and each individual point. The method used to accomplish this is called the "least squares" method. The indicator is based on the theory that prices are pulled back to the regression line after they stray above it or below it.

The Linear Regression Indicator can also be used to monitor the current trend. If it's rising the trend is up, and if it's falling the trend is down. For our Linear Regression and Momentum System, we'll smooth the linear regression line with an exponential moving average so that the indicator does not switch back and forth between uptrends and downtrends too often.

Momentum, the second indicator in this system, compares the current bar's closing price with the closing price a specified number of bars in the past. To calculate a 5-bar momentum line, for example, subtract the close of 5 bars ago from the current bar's close.

When the 5-bar Momentum Indicator is above its zero line and rising, the 5-bar price changes are positive and increasing — that is, the trend is bullish and accelerating. If the momentum line turns flat, it implies that the 5-bar price changes are about equal during the Momentum Indicator's period of sideways movement. When the Momentum Indicator begins to decline from above zero, the market's gains during the past 5 bars are less than the corresponding gains in the preceding bars — that is, the uptrend is decelerating.

When the 5-bar Momentum Indicator falls below its zero line, the current close is below the close 5 bars ago. As the downtrend gains bearish velocity, momentum accelerates downward from the zero line. An upturn of the indicator in negative territory means that the magnitude of 5-bar price declines is decreasing — that is, the downtrend is decelerating. Momentum is a leading indicator — it levels off while prices are still rising in an uptrend or falling in a downtrend, and it reverses its direction when the trend begins to slow.

In this system, we'll use momentum to identify countertrend declines in an uptrend and countertrend rallies in a downtrend. As mentioned previously, the trend will be determined by the direction of a smoothed linear regression line (i.e., an exponential moving average of linear regression).

Figure 7 shows a daily chart of AT&T with a linear regression line and Momentum Indicator applied.

Figure 7. Daily AT&T chart with the smoothed linear regression line and Momentum Indicator applied

For a buy setup, we'll require the smoothed linear regression line to be rising and for momentum to be below zero but rising. For a sell setup, we'll require the smoothed linear regression line to be falling and for momentum to be above zero but falling.

After a buy setup, we'll calculate the distance between the high of the setup bar and the smoothed linear regression line. Then we'll add 50% of that distance to the high of the setup bar to determine our long entry price.

After a sell setup, we'll calculate the distance between the low of the setup bar and the smoothed linear regression line. Then we'll subtract 50% of that distance from the low of the setup bar to determine our short entry price.

Once we've entered a long position, we'll set our initial protective stop at the low of the setup bar minus 50% of the 10-bar average true range; once we've entered a short position, we'll set our initial protective stop at the high of the setup bar plus 50% of the 10-bar average true range. As the market moves in our favor, we'll trail a % risk trailing stop.

To exit when we're in a long position, we'll watch for the smoothed linear regression line to turn down. Then we'll place our sell stop at the lowest low of the last 4 bars. When we're in a short position, we'll watch for the smoothed linear regression line to turn up. Then we'll place our buy stop at the highest high of the last 4 bars.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some setup work to calculate the linear regression line, its smoothed average and momentum. The setup, entries and exits are described next:

Setup

a) Calculate the linear regression line (using the closing prices and a length of 20) and its exponential average (using a length of 15).

b) Calculate the momentum (using the closing prices and a length of 10).

Long Entries

a) For a buy setup, we'll require the smoothed linear regression line to be rising. In other words, its value on the current bar must be greater than its value on the previous bar. Also, momentum must be below zero but rising (greater on this bar than on the previous bar).

b) After a buy setup, we'll calculate the distance between the high of the setup bar and the smoothed linear regression line. Then we'll add 50% of that distance to the high of the setup bar; this is our long entry price. The order remains active for 4 bars.

Short Entries

a) For a sell setup, we'll require the smoothed linear regression line to be falling. In other words, its value on the current bar must be less than its value on the previous bar. Also, momentum must be greater than zero but falling (less on this bar than on the previous bar).

b) After a sell setup, we'll calculate the distance between the low of the setup bar and the smoothed linear regression line. Then we'll subtract 50% of that distance from the low of the setup bar; this is our short entry price. The order remains active for 4 bars.

Exit Orders

a) Once we've entered a long position, we'll set our initial protective stop at the low of the setup bar minus 50% of the 10-bar average true range.

b) Once we've entered a short position, we'll set our initial protective stop at the high of the setup bar plus 50% of the 10-bar average true range.

d) To exit when we're in a long position, we'll watch for the smoothed linear regression line to turn down (the value on the current bar is less than it was on the previous bar). Then we'll place a stop order at the lowest low of the last 4 bars.

e) When we're in a short position, we'll watch for the smoothed linear regression line to turn up (the value on the current bar is greater than it was on the previous bar). Then we'll place a stop order at the highest high of the last 4 bars.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Linear Regression & Momentum (STAD 4: LinReg/Mom)

```
Inputs: Price(Close), LRLen(20), AvgLen(15), MomLen(10), Pcnt(.5), Trade(10000), Lot(100);
Vars: XLinReg(0), Mom(0), SetBarH(0), SetBarL(0), SetBarS(0), CountL(0), CountS(0);

{Assignment of Exponential Linear Regression and Momentum}
XLinReg = XAverage(LinearRegValue(Price, LRLen, 0), AvgLen);
Mom = Momentum(Price, MomLen);

{Accumulates to count the bars after the SetUps below}
CountL = CountL + 1;
CountS = CountS + 1;

{Assignment of System Entry/Exit criteria}
Condition1 = XLinReg > XLinReg[1];
Condition2 = Mom < 0 AND Mom > Mom[1];
Condition3 = XLinReg < XLinReg[1];
Condition4 = Mom > 0 AND Mom < Mom[1];

{Check criteria and generate Buy order if criteria have been met within 4 bars}
If Condition1 AND Condition2 then Begin
        If CountL = 1 OR CountL > 4 then Begin
                SetBarL = High;
                CountL = 1;
        End;

        If CountL <= 4 then Begin
                Value1 = Pcnt * (SetBarL - XLinReg);
                Buy ("Long") NumUnits(Trade,Lot) Shares Next Bar at SetBarL + Value1 Stop;
        End;
End;

{Check criteria and generate Sell order if criteria have been met within 4 bars}
If Condition3 AND Condition4 then Begin
        If CountS = 1 OR CountS > 4 then Begin
                SetBarS = Low;
                CountS = 1;
        End;
        If CountS <= 4 then Begin
                Value1 = Pcnt * (SetBarS - XLinReg);
                Sell ("Short") NumUnits(Trade,Lot) Shares Next Bar at SetBarS - Value1 Stop;
        End;
End;

{Initial Stops}
ExitLong ("X")Next Bar from entry ("Long") AT$ Low - (Pcnt * AvgTrueRange(10)) Stop;
ExitShort ("Y")Next Bar from entry ("Short") AT$ High + (Pcnt * AvgTrueRange(10)) Stop;
```

```
{Trailing Stops}
If Condition3 AND Low < Lowest(Low, 4)[1] then
        ExitLong Next Bar at Market;

If Condition1 AND High > Highest(High, 4)[1] then
        ExitShort Next Bar at Market;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default	Description
Price	Close	Price used to calculate the linear regression line.
LRLen	20	Length, expressed in bars, used to calculate the linear regression line.
AvgLen	1	Length, expressed in bars, used to calculate the exponential average of the linear regression line.
MomLen	10	Length, expressed in bars, to use to calculate the momentum.
Pcnt	.5	Percentage used to calculate the value of the entries, exits and stops.
Trade	10,000	Dollar amount willing to invest in each transaction
Lot	100	Minimum lot size for each transaction

In addition to these inputs, we define the following variables:

```
Vars: XLinReg(0), Mom(0), SetBarH(0), SetBarL(0), SetBarS(0), CountL(0), CountS(0);
```

Setup

First, we calculate the exponential moving average of the linear regression. We use the LinearRegValue function to calculate the linear regression and then use the resulting value as an input for the XAverage function in order to determine the exponential average of the linear regression. We store the resulting value in the XLinReg variable.

We also calculate the momentum using the Momentum function and store the resulting value in the Mom variable.

```
XLinReg = XAverage(LinearRegValue(Price, LRLen, 0), AvgLen);
Mom = Momentum(Price, MomLen);
```

We use two counters, CountL and CountS. We increment these with each bar and reset them each time a buy setup or sell setup occurs, respectively. This way, we can keep a count and use them to keep buy and sell orders active for 4 bars:

```
CountL = CountL + 1;
CountS = CountS + 1;
```

Next, we make the comparisons between the exponential moving averages and the momentum values. The results of these comparisons are used throughout the system. We compare the exponential moving average of the linear regression line to the same value one bar ago. We also compare the value of the momentum calculation to zero and to the same value one bar ago.

First, we check to see if the current value of the exponential average is greater than the value one bar ago. If it is, the variable Condition1 is set to True. Next, we check to see if the momentum value is less than zero and greater than the same value one bar ago. If it is, Condition2 is set to

True. After that, we check to see if the exponential average is less than it was one bar ago. If it is, Condition3 is set to True. And finally, we check to see if the momentum value is greater than zero and less than the same value one bar ago. If it is, Condition4 is set to True.

```
Condition1 = XLinReg > XLinReg[1];
Condition2 = Mom < 0 AND Mom > Mom[1];
Condition3 = XLinReg < XLinReg[1];
Condition4 = Mom > 0 AND Mom < Mom[1];
```

Long Entries

Next, we check to see if Condition1 and Condition2 are true. If they are, and they became true either on the previous bar or more than 4 bars ago, then the high price is stored in the variable SetBarL and the CountL variable, which accumulates the number of bars since the setup, is set to 1.

```
If Condition1 AND Condition2 then Begin
        If CountL = 1 OR CountL > 4 then Begin
                SetBarL = High;
                CountL = 1;
        End;
```

If the number of bars since Condition1 and Condition2 became true is less than 4, then we place a buy order at a certain percentage above the value stored in SetBarL. We calculate the percentage by subtracting the value of the exponential moving average from the value in SetBarL and multiplying that by the specified percentage (default of .5). We add this to the value in SetBarL and that is our entry price. Notice that we specified a certain number of shares to buy using the NumUnits function. This function is described in detail in the section at the end of this chapter, titled, “Investing a Fixed Dollar Amount.”

```
        If CountL <= 4 then Begin
                Value1 = Pcnt * (SetBarL - XLinReg);
                Buy ("Long") NumUnits(Trade,Lot) Shares Next Bar at SetBarL+
                        Value1 Stop;
        End;
End;
```

Short Entries

Next, we check to see if Condition3 and Condition4 are true. If they are, and they became true either on the previous bar or more than 4 bars ago, then the low price is stored in the variable SetBarS and the CountS variable, which accumulates the number of bars since the setup, is set to 1.

```
If Condition3 AND Condition4 then Begin
        If CountS = 1 OR CountS > 4 then Begin
                SetBarS = Low;
                CountS = 1;
        End;
```

If the number of bars since Condition3 and Condition4 became true is less than 4, then we place a sell order at a certain percentage below the value stored in SetBarS. We calculate the percentage by subtracting the value of the exponential moving average from the value in SetBarS and multiplying that by the specified percentage (default of .5). We subtract this from the value in SetBarS and that is our entry price. Notice that we specified a certain number of shares to sell using the NumUnits function. This function is described in detail in the section at the end of this chapter, titled, “Investing a Fixed Dollar Amount.”

```
If CountS <= 4 then Begin
                Value1 = Pcnt * (SetBarS - XLinReg);
                Sell ("Short") NumUnits(Trade,Lot) Shares Next Bar at SetBarS -
```

```
                    Value1 Stop;
                End;
            End;
```

Long & Short Exits

We will exit any long position at or below the value that results when you subtract a percentage (as specified by the input Pcnt) of the 10-bar average of the true range from the low of the buy setup bar.

Conversely, we will exit the short position at or above the value that results when you add a percentage (as specified by the input Pcnt) of the 10-bar average of the true range to the high of the sell setup bar.

Notice the use of the Reserved Word AT$. This ties the following price reference to the bar of entry. It is a simple yet powerful technique to use.

```
ExitLong ("X")Next Bar from entry ("Long") AT$ Low - (Pcnt * AvgTrueRange(10)) Stop;
ExitShort ("Y")Next Bar from entry ("Short") AT$ High + (Pcnt * AvgTrueRange(10)) Stop;
```

Finally, we will place a trailing stop for long positions at the lowest low of the last four bars. Conversely, we will place a trailing stop to cover our short positions at the highest high of the last four bars.

```
If Condition3 AND Low < Lowest(Low, 4)[1] thenExitLong Next Bar at Market;
If Condition1 AND High > Highest(High, 4)[1] thenExitShort Next Bar at Market;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified $10,000 as the trade size and 100 shares as the lot size.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

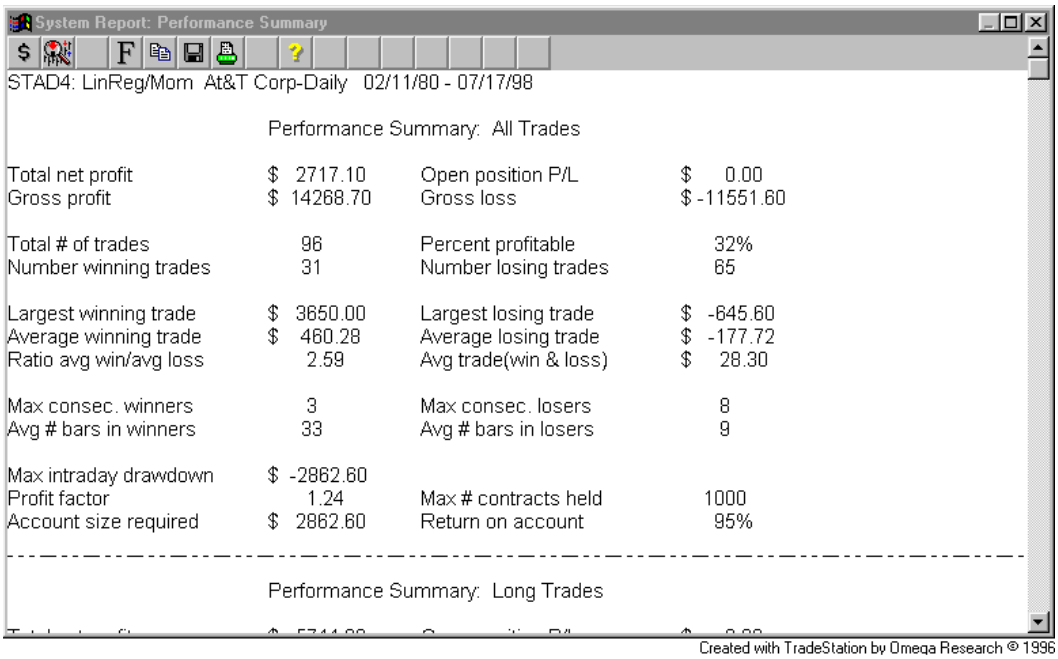
Under the **Stops** tab, we enabled a % risk trailing stop. This type of stop enables you to indicate what percent of the maximum open profit you are willing to give back before the position is automatically closed out. The % risk trailing stop also requires that you specify a minimum profit level that must be reached before the stop takes effect.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the **Do not allow multiple entries in the same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

A winning percentage of 32 coupled with a ratio of average win to average loss of 2.59 is marginally acceptable. Figure 8 shows the sample System Report for the Linear Regression &Momentum System on a daily AT&T chart.


System Report: Performance Summary

STAD4: LinReg/Mom At&T Corp-Daily 02/11/80 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 2717.10	Open position P/L	$ 0.00
Gross profit	$ 14268.70	Gross loss	$ -11551.60
Total # of trades	96	Percent profitable	32%
Number winning trades	31	Number losing trades	65
Largest winning trade	$ 3650.00	Largest losing trade	$ -645.60
Average winning trade	$ 460.28	Average losing trade	$ -177.72
Ratio avg win/avg loss	2.59	Avg trade(win & loss)	$ 28.30
Max consec. winners	3	Max consec. losers	8
Avg # bars in winners	33	Avg # bars in losers	9
Max intraday drawdown	$ -2862.60		
Profit factor	1.24	Max # contracts held	1000
Account size required	$ 2862.60	Return on account	95%

Performance Summary: Long Trades

Created with TradeStation by Omega Research © 1996

igure 8. Sample System Report for the Linear Regression & Momentum System applied to a daily At&T chart

The system made $1.24 (the profit factor) for each dollar it lost. Unlike many of our systems for stocks, this one actually made some money on the short side as well as the long side. The Performance Summary also shows that the system stayed in winners an average of 33 bars (letting profits run) but stayed in losers for only nine bars (cutting losses short).

Suggestion for Improvement

As we scrolled through the years of AT&T daily data, we noticed that many of the buy signals that turned out to be losses occurred when the smoothed linear regression line was rising (as required in this system), but when prices were below the line. Similarly, many of the signals to sell short that ended up losers occurred when the smoothed linear regression line was falling (as required), but when prices were above the line.

We would like to see what happens to this system's performance when we strengthen the entry requirement to include that prices must be above the smoothed linear regressions line for a buy and below the line for a sell. (Note in Figure 7 that both profitable long entries were executed above the smoothed linear regression line.)

Herrick Payoff Index & Channel Breakout on Close System

The Herrick Payoff Index (HPI) is a complex formula that evaluates changes in price, volume, and open interest. HPI measures the flow of money into and out of a market.

Because its calculation includes open interest, HPI can only be used for futures and not for stocks. Another restriction of the HPI is that it can only be applied to daily data because there is no such thing as intraday or weekly open interest.

HPI can be used to confirm price action. When HPI rallies to a new high along with a new price high, volume (the number of contracts that change hands during a specified time period) and

open interest are confirming the bullish price action, and when HPI falls to a new low along with a new low in price, volume and open interest are confirming the bearish price action.

What is Open Interest? Open interest is the number of futures contracts outstanding at the end of a trading day.

Open interest increases or decreases as traders take new positions in a market or exit from old positions.

Open interest increases when a buyer and a seller take new positions in a market. Open interest decreases when a trader who is long and a trader who is short make a trade to exit their positions.

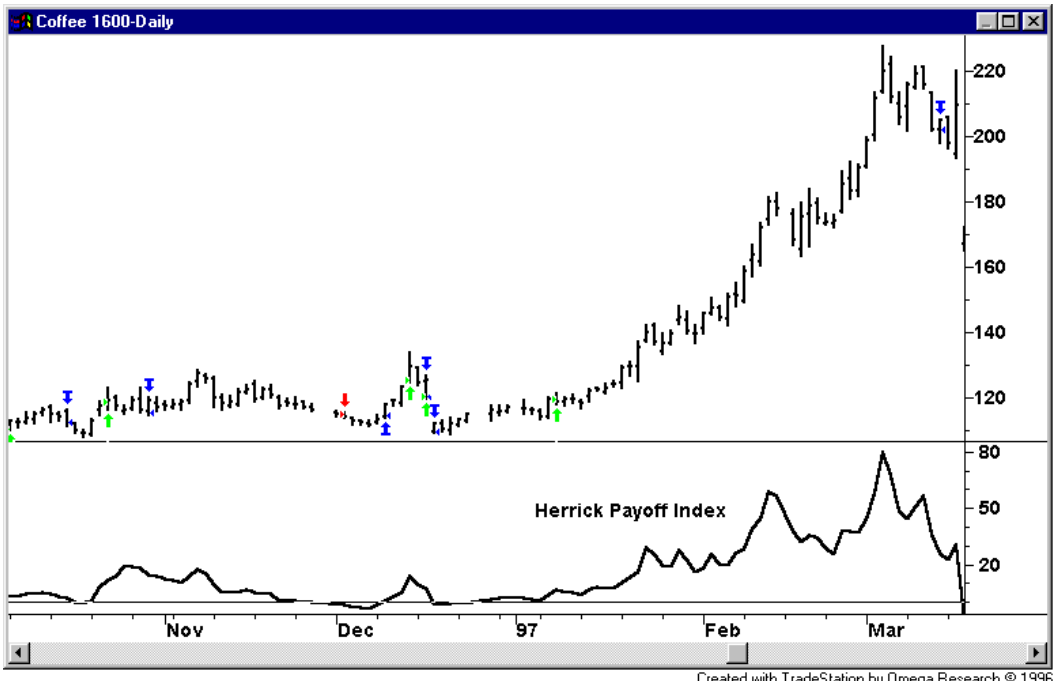
However, when a new price high is accompanied by the failure of HPI to also make a new high, the price action is not confirmed by volume and open interest. A bearish divergence exists between the higher price high and the lower HPI high. Similarly, when a new price low is not matched by a new HPI low, the price action is unconfirmed by volume and open interest, and a condition of bullish divergence exists.

Bullish and bearish divergences are early warning signs of a possible change in trend. HPI confirmation of new price highs and new price lows suggests that the current trend is more likely to continue than to reverse.

Another component of this system is the channel breakout on close. When a market closes above the highest high of a specified number of bars, price action is bullish, and when a market closes below the lowest low of a specified number of bars, the market is bearish.

A system combining the Channel Breakout on Close Indicator and the Herrick Payoff Index should alert us to good trading opportunities. After a bullish setup, we'll enter long at the high of the setup bar plus one point, and after a bearish setup, we'll enter short at the low of the setup bar minus one point. The setup remains in effect for five days after the setup bar.

Figure 9 shows a daily Coffee chart with this system applied along with the HPI Indicator (built into TradeStation).



igure 9. Herrick Payoff Index applied to a daily Coffee chart

Our initial stop and trailing stop when long will be at the four-day low, and our initial stop and trailing stop when short will be at the four-day high. We'll exit long positions on the close when HPI crosses below zero and exit short positions on the close when HPI crosses above zero.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some setup work to calculate the Channel Breakout on Close Indicator and the Herrick Payoff Index, as well as their to compare them. The setup, entries and exits are described next.

Setup

a) Calculate the HPI values.

b) Look for confirmation of bearish price action by consistency between the closing price and the HPI. The closing price will be lower than the lowest low of the last 10 bars and the HPI will be lower than the lowest HPI of the last 10 bars. This is our sell setup.

c) Look for confirmation of bullish price action by consistency between the closing price and the HPI. The closing price will be higher than the highest high of the last 10 bars and the HPI will be higher than the highest HPI of the last 10 bars. This is our buy setup.

Long Entries

a) After a bullish or buy setup, we'll enter long at the high of the setup bar plus one point.This setup will remain in effect for five days after the setup bar.

Short Entries

b) After a bearish or sell setup, we'll enter short at the low of the setup bar minus one point. This setup will remain in effect for five days after the setup bar.

Exit Orders

a) Our initial stop and trailing stop when long will be at the 4-day low, and our initial stop and trailing stop when short will be at the 4-day high.

b) We'll also exit long positions on the close when HPI crosses below zero and exit short positions on the close when HPI crosses above zero.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Herrick Payoff Index & Channel Breakout on Close (STAD 4: Herrick/Break)

```
Inputs: Mult(100), Factor(.1), Length(10);
Vars: HPIVal(0), CountL(5), CountS(5);

{HPI calculation and System criteria assignments to variables}
HPIVal = HPI(Mult, Factor);
Condition1 = Close > Highest(High, Length)[1];
Condition2 = HPIVal > Highest(HPIVal, Length)[1] AND HPIVal > 0;
Condition3 = Close < Lowest(Low, Length)[1];
Condition4 = HPIVal < Lowest(HPIVal, Length)[1] AND HPIVal < 0;

{Accumulates to count the bars after the SetUps below}
CountL = CountL + 1;
CountS = CountS + 1;

{System Entry SetUps}
If Condition1 AND Condition2 AND CountL > 5 thenCountL = 1;
If Condition3 AND Condition4 AND CountS > 5 thenCountS = 1;

{System Entries if they are within 5 bars of the setups}
IF CountL <= 5 Then
        Buy Next Bar at High[CountL] + 1 Point Stop;
```

```
IF CountS <= 5 Then
        Sell Next Bar at Low[CountS] - 1 Point Stop;

{Trailing Stops}
ExitLong Next Bar at Lowest(Low, 4) Stop;
ExitShort Next Bar at Highest(High, 4) Stop;

{System Stops}
IF HPIVal < 0 ThenExitLong This Bar on Close;
IF HPIVal > 0 ThenExitShort This Bar on Close;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default	Description
Mult	100	The contract value of a 1 cent move in the underlying asset.
Factor	.1	Smoothing factor, used to smooth the HPI.
Length	10	Length, expressed in bars, used to calculate the x-bar highs and lows as well as the Herrick Payoff Index.

In addition to these inputs, we define the following variables (notice that our variables CountL and CountS are initialized to 5):

```
Vars: HPIVal(0), CountL(5), CountS(5);
```

Setup

We begin by calculating the HPI value using the HPI function and a contract value of 100 and a smoothing factor of .1. We store the resulting value in the variable HPIVal.

```
HPIVal = HPI(Mult, Factor);
```

Then we perform four comparisons and store the results in four different variables, Condition1 through Condition4. First, we check to see if the close of the current bar is greater than the highest high of the last 10 bars. We store the result in Condition1. Then we check to make sure the value in HPIVal is greater than the highest HPIVal value for the last 10 bars and make sure HPIVal is greater than zero. The result is stored in Condition2.

Next we check to see if the close of the current bar is less than the lowest low of the last 10 bars. We store the result in Condition3. And finally, we check to see if HPIVal is less than the lowest HPIVal value for the last 10 bars and make sure HPIVal is less than zero. The result is stored in Condition4.

```
Condition1 = Close > Highest(High, Length)[1];
Condition2 = HPIVal > Highest(HPIVal, Length)[1] AND HPIVal > 0;
Condition3 = Close < Lowest(Low, Length)[1];
Condition4 = HPIVal < Lowest(HPIVal, Length)[1] AND HPIVal < 0;
```

We use two counters, CountL and CountS. We increment these with each bar and reset them to 1 each time a buy setup or sell setup occurs, respectively. This way, we can keep a count and use them to keep buy and sell orders active for 5 bars:

```
CountL = CountL + 1;
CountS = CountS + 1;
```

If the criteria specified in the conditional statements above, as appropriate, are true and the variables CountL or CountS is greater than 5, then CountL or CountS is reset to one. Since the buy or sell setup criteria is true, we want to place buy or sell orders, respectively, therefore, we need to set the counters to 1 so the next set of instructions are carried out.

```
If Condition1 AND Condition2 AND CountL > 5 thenCountL = 1;
If Condition3 AND Condition4 AND CountS > 5 thenCountS = 1;
```

Long & Short Entries

If the CountL variable is less than or equal to 5, a buy stop order is generated at the high of the setup bar plus 1 point. If the CountS variable is less than or equal to 5, a sell stop order is generated at the low of the setup bar minus 1 point.

```
If CountL <= 5 then Buy Next Bar at High[CountL] + 1 Point Stop;
If CountS <= 5 thenSell Next Bar at Low[CountS] - 1 Point Stop;
```

Long & Short Exits

A trailing stop for long positions will be placed at the lowest low of the last four bars. Conversely, we will place a trailing stop to cover our short positions at the highest high of the last four bars.

```
ExitLong Next Bar at Lowest(Low, 4) Stop;
ExitShort Next Bar at Highest(High, 4) Stop;
```

Additionally, the system will generate a long exit on the close of the current bar if the HPI value falls below zero. A short exit on the close of the current bar will be generated if the HPI value rises above zero.

```
If HPIVal < 0 thenExitLong this bar on Close;
If HPIVal > 0 then ExitShort this bar on Close;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified 1 as the default contract size to trade.

Note: *Remember that Commissions are calculated on a per contract basis. When you are trading commodities, you would enter the average commission you are charged divided by the number of contracts the system is buying and selling.*

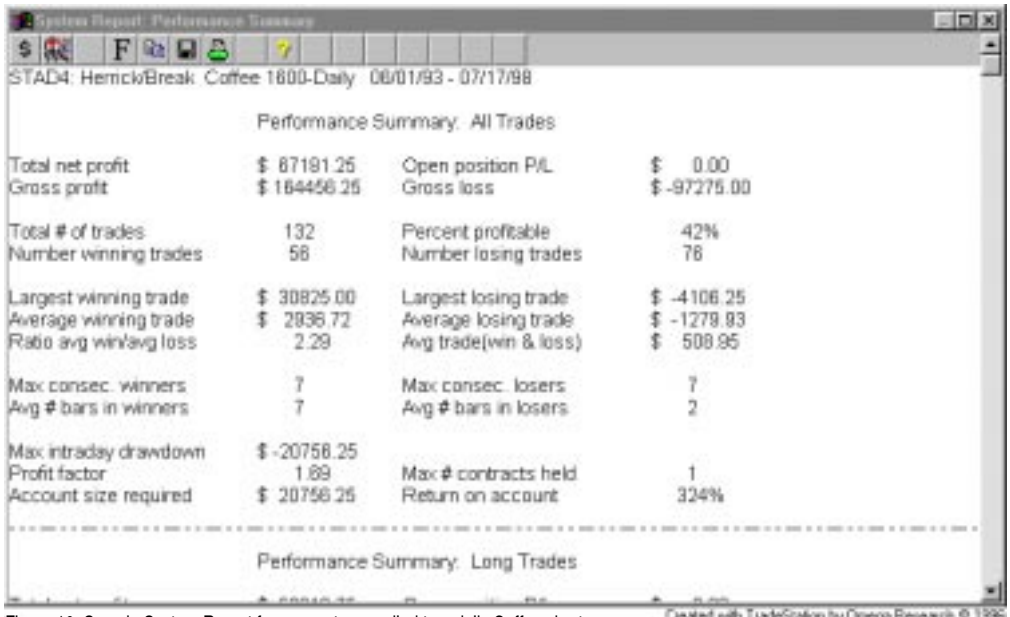
Under the **Stops** tab, we enabled a money management stop. Our trailing stop for this system will be at the four-bar low when long and at the four bar high when short.

Note: *When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the **Do not allow multiple entries in the same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

The Performance Summary in Figure 10 shows a large net profit of over $67,000 per contract.



STAD4: Herrick/Break: Coffee 1600-Daily 08/01/93 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 67191.25	Open position P/L	$ 0.00
Gross profit	$ 164456.25	Gross loss	$ -97275.00
Total # of trades	132	Percent profitable	42%
Number winning trades	56	Number losing trades	76
Largest winning trade	$ 30825.00	Largest losing trade	$ -4106.25
Average winning trade	$ 2936.72	Average losing trade	$ -1279.93
Ratio avg win/avg loss	2.29	Avg trade(win & loss)	$ 508.95
Max consec. winners	7	Max consec. losers	7
Avg # bars in winners	7	Avg # bars in losers	2
Max intraday drawdown	$ -20756.25		
Profit factor	1.69	Max # contracts held	1
Account size required	$ 20756.25	Return on account	324%

Performance Summary: Long Trades

Created with TradeStation by Omega Research © 1996

Figure 10. Sample System Report for our system applied to a daily Coffee chart

The other performance numbers are good also with 42% profitable trades, a ratio of average win to average loss of 2.29, and a profit factor of 1.69. Maximum drawdown is high at $20,756.25, or 30% of the total net profit.

An even more serious potential problem is that the largest winning trade ($30,825) comprises such a big percentage (46%) of the total net profit. If we were to subtract the largest winning trade from this system's track record, would the results still encourage us to trade the system? In this extreme example, no.

We optimized for the best number of bars to include in the Channel Breakout on Close Indicator, testing the values from 2 through 8 in increments of one. Figure 11 shows the resulting Optimization Report.

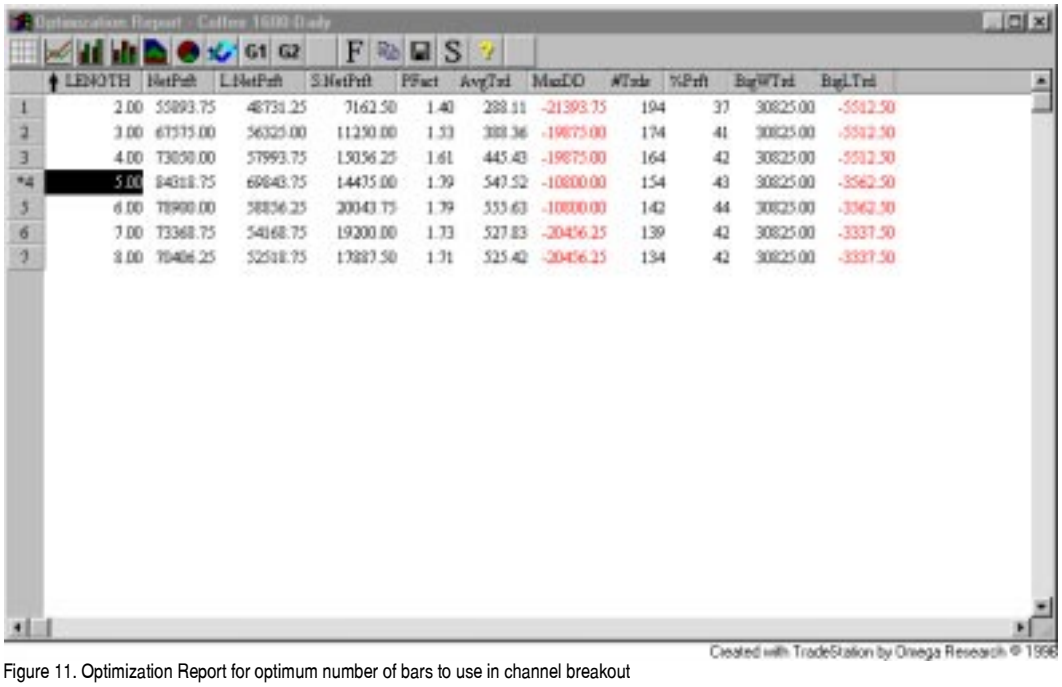


Figure 11. Optimization Report for optimum number of bars to use in channel breakout

The 5-bar breakout yielded the best results with the largest net profit and profit factor (tied with the six-bar breakout), and the second best average trade. The bar graph of the net profits is picture-perfect, as shown in Figure 12, with the optimal value of five in the center of the range of values, and descending profits for the values to its left and right.

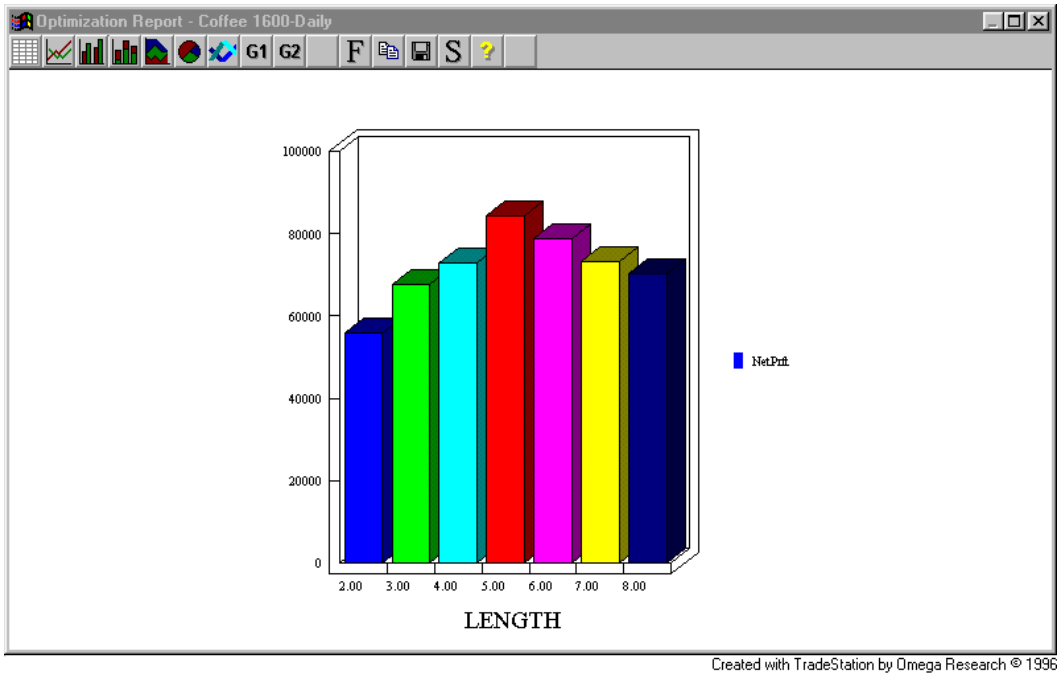


Figure 12. Bar graph view of the above Optimization Report

Suggestions for Improvement

The System Equity Indicator in Figure 13 shows how profits in the Coffee market soared during the major uptrend.

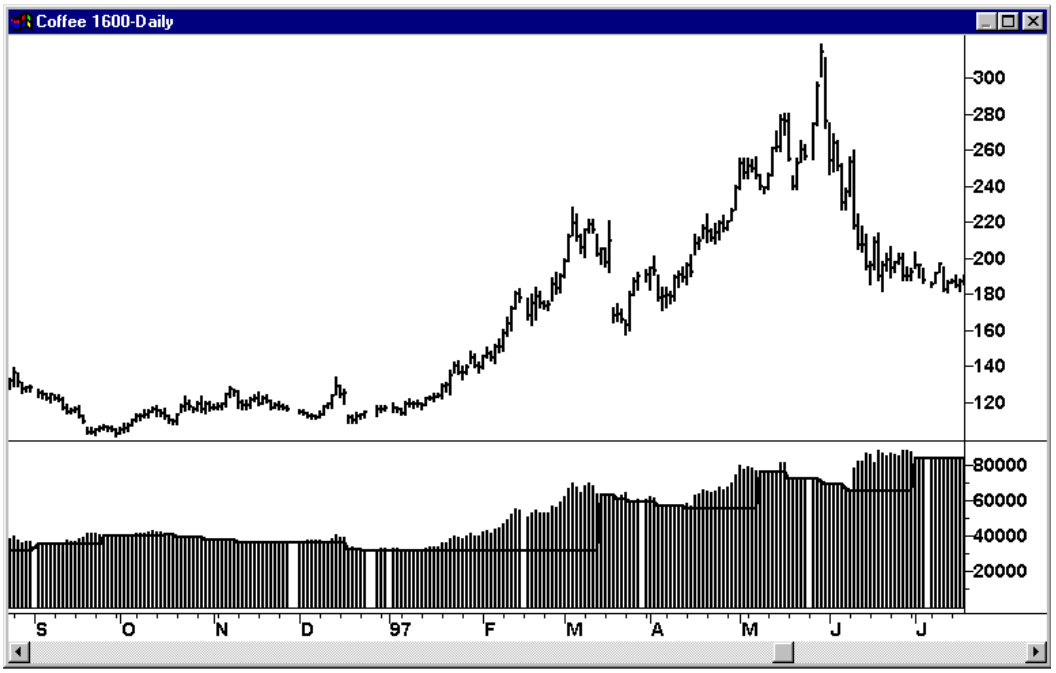


Figure 13. System Equity Indicator applied to the chart

Since we wouldn't really be excited about trading the system without that one great trade, we know that we'll have to improve this system before we'll have confidence in it.

One possibility for improvement has more to do with the markets we trade than with this particular system. Perhaps we could apply a "trendiness" indicator such as the ADX (Average Directional Index) to a list of the commodities we are thinking about trading and selecting only those that historically have displayed the greatest tendency to trend.

Consecutive Close System

To try something different from most of our previous STAD Club systems, we decided to write a system with an entry based on a very simple chart pattern and an exit based on a profit target rather than on the reversal of an indicator.

The chart pattern for a long setup begins with at least three consecutive up closes. This condition should tell us that the short-term trend is up. However, three consecutive up closes may also suggest that the market may be a bit overextended and due for a correction. Therefore, after three consecutive up closes, we'll wait for a down close. Our idea is that the down close should alleviate the market's short-term overbought condition and complete the setup.

Therefore, our buy setup is the three up close/one down close pattern and we'll place a long entry when the price moves above the open of the bar that follows our setup. We'll enter long on the bar after the setup at the open plus 25% of the 8-bar average true range.

Once we initiate a new long position, we'll place our protective stop at the 10-bar low. Then, as the market trades higher, we'll trail a % risk trailing stop.

Our exit for the Consecutive Close System is unusual. We'll exit at the open of the next bar once two bars have closed above the high of the three up close/one down close pattern. In other words, after we buy, we'll wait for the market to post two opens above the high of the setup

pattern, and we'll exit on the next open. Our thinking behind this exit strategy is that since the setup and entry are based on short-term price action, the exit should be as well. Figure 14 shows the chart pattern we look for to initiatie a long position:

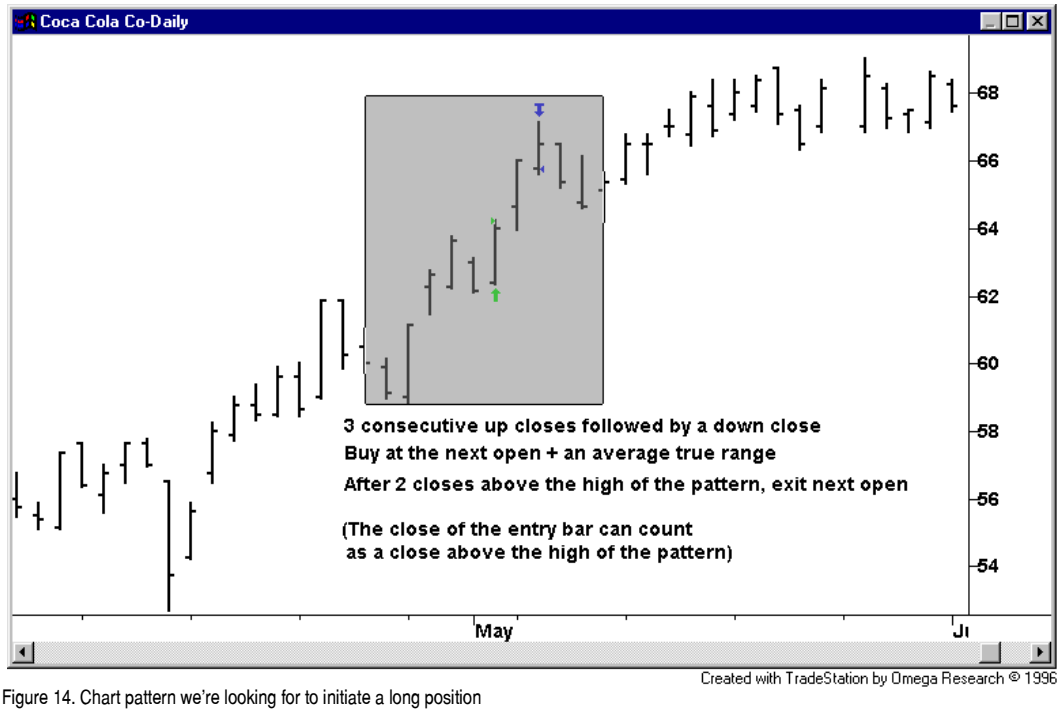


Figure 14. Chart pattern we're looking for to initiate a long position

Let's take a look at the short side of our system. The chart pattern for a short setup begins with at least three consecutive down closes. This condition indicates that the short-term trend is down. However, three consecutive down closes may also indicate that the market is susceptible to a correction. Therefore, after three consecutive down closes, we'll wait for an up close. The up close should cancel the market's short-term oversold condition.

Therefore, our sell setup is the three down close/one up close pattern and we'll place a short entry when the price moves below the open of the bar that follows the setup. We'll enter the market short on the bar after the setup at the open minus 25% of the 8-bar average true range.

After we take a short position, we'll set our protective stop at the 10-bar high. Then, as the market trades lower, we'll trail a % risk trailing stop. We'll exit on the open after the second

consecutive close below the low of the setup pattern. Figure 15 illustrates the short side of the system:

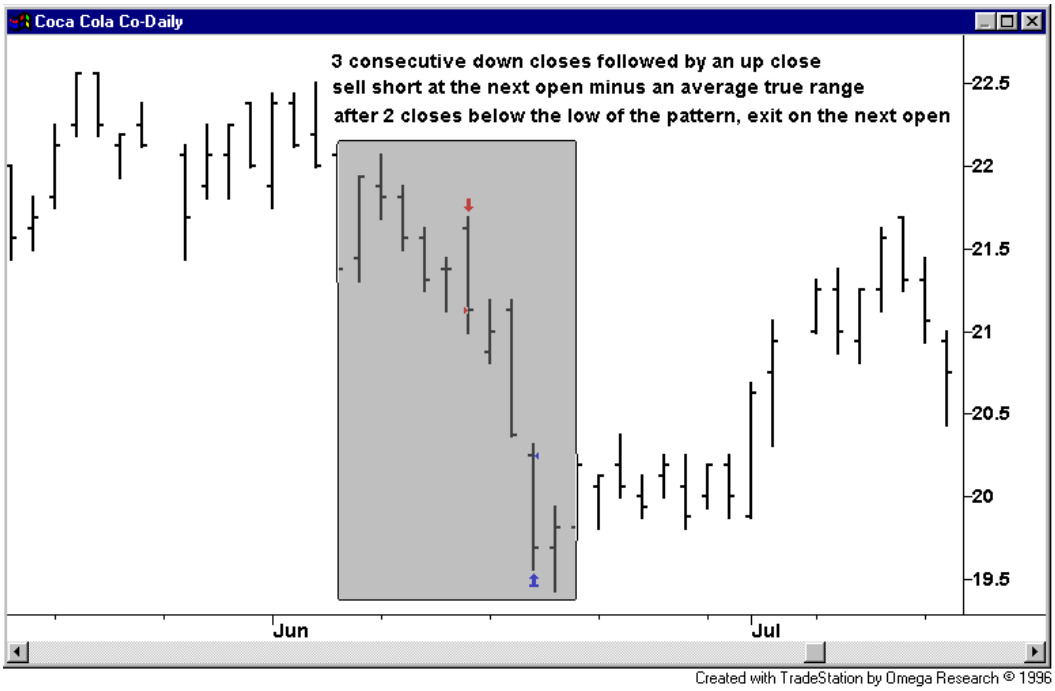


Figure 15. Chart pattern we're looking for to initiate a short position

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. The entries and exits are described next:

Long Entries

a) We look for three up closes and a subsequent down close. This is our buy setup. We place a buy stop order on the bar after the setup at the open plus 25% of the 8-bar average true range.

Short Entries

a) We look for three consecutive down closes and a subsequent up close. This is our sell setup. We place a sell stop order on the bar after the setup at the open minus 25% of the 8-bar average true range.

Exit Orders

a) Once we initiate a position, we'll place our protective stop at the 10-bar low and 10-bar high for long and short positions, respectively.

b) After we buy, we'll wait for the market to post two opens above the high of the setup pattern, and we'll exit on the subsequent open.

c) After we take a short position, we'll wait for the market to post two closes below the low of the setup pattern, and we'll exit on the subsequent open.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Consecutive Close (STAD 4: Close Pattern)

```
Input: NumUpBar(3), NumDnBar(3), AvgLen(8), Pcnt(.25);
Vars: HH(0), LL(0), LCount(0), SCount(0);

{Identifying the consecutive close pattern and placing a buy order}
If MRO( Close < Close[1], NumUpBar, 1)[1] = -1 and Close < Close[1] then Begin
        Buy at Open next bar + Average(TrueRange, AvgLen)*Pcnt Stop;
        HH = Highest(High, NumUpBar + 1);
End;

{Identifying the consecutive close pattern and placing a sell order}
If MRO( Close > Close[1], NumDnBar, 1)[1] = -1 and Close>Close[1] then Begin
        Sell at Open next bar - Average(TrueRange, AvgLen)*Pcnt Stop;
        LL = Lowest(Low, NumDnBar + 1);
End;

{Exit criteria based on short term price action}
If MarketPosition <> 1 thenLCount = 0;
If MarketPosition <> -1 thenSCount = 0;

If MarketPosition = 1 and Open next bar > HH then Begin
        LCount = LCount + 1;
        If LCount = 2 then ExitLong next bar at Open;
End;

If MarketPosition = -1 and Close < LL then Begin
        SCount = SCount + 1;
        If SCount = 2 then ExitShort next bar at Open;
End;

{Protective stops}
ExitShort next bar at Highest(High,10) Stop;
ExitLong next bar at Lowest(Low,10) Stop;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default	Description
NumUpBar	3	Number of up closes that are required for a buy setup.
NumDnBar	3	Number of down closes that are required for a sell setup.
AvgLen	8	Length, expressed in bars, used to calculate the average true range.
Pcnt	.25	Percentage used to calculate the value of the entries and exits.

In addition to these inputs, we define the following variables:

```
Vars: HH(0), LL(0), LCount(0), SCount(0);
```

Long Entries

We used the Most Recent Occurrence (MRO) function is used to determine if there were any down closes in the last 3 bars, not including the current bar. The function returns the number of

the bar on which the down close most recently occurred. If no down closes are found, the MRO function returns a -1.

Therefore, if a down close has not occurred in the last 3 bars, and the current close is greater than the close of the previous bar, we place a buy stop order at the open of the next bar plus a percentage 2% of the 8-bar average of the true range. Also, we set the variable HH to the highest high of the last 4 bars. We'll use the value in HH in our exit criteria.

```
If MRO( Close < Close[1], NumUpBar, 1)[1] = -1 and Close<Close[1] then Begin
        Buy at Open next bar + Average(TrueRange, AvgLen) * Pcnt Stop;
        HH = Highest(High, NumUpBar + 1);
End;
```

Short Entries

Like with the long entries, we again use the MRO function to determine if there have been any up closes in the last 4 bars. If an up close has not occurred and the current close is less than the close of the previous bar, we place a sell stop at the open of the next bar minus 25% of the 8-bar average of the true range. Also, the lowest low of the last 4 bars is stored in the variable LL. We'll use the value in LL in our exit criteria.

```
If MRO( Close > Close[1], NumDnBar, 1)[1] = -1 and Close>Close[1] then Begin
        Sell at Open next bar - Average(TrueRange, AvgLen)*Pcnt stop;
        LL = Lowest(Low, NumDnBar + 1);
End;
```

Long & Short Exits

If the current market position is short or flat, the variable LCount is re-set to 0.

```
If MarketPosition <> 1 then LCount = 0;
```

If the current market position is long or flat, the variable SCount is re-set to 0.

```
If MarketPosition <> -1 then SCount = 0;
```

If the current market position is long and the next bar's open is greater than the value stored in HH, then the LCount variable is accumulated by one. If these conditions exist and LCount holds a value of 2, meaning that the open has been higher than the value in HH for two bars now, we exit our long position at the open of the next bar.

```
If MarketPosition = 1 and Open next bar > HH then Begin
        LCount = LCount + 1;
        If LCount = 2 thenExitLong next bar at open;
End;
```

If the current market position is short and the next bar's close is less than the value stored in LL, then the SCount variable is accumulated by one. If these conditions exist and SCount holds a value of 2, meaning that the close has been less than the value in LL for two bars now, we exit our short position at the open of the next bar.

```
If MarketPosition = -1 and Close < LL then Begin
        SCount = SCount + 1;
        If SCount = 2 then
                ExitShort next bar at Open;
End;
```

Finally, we will use the highest high and lowest low of the last 10 bars as a trailing stop.

```
ExitShort next bar at Highest(High,10) Stop;
ExitLong next bar at Lowest(Low,10) Stop;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified $10,000 as the trade size and 100 shares as the lot size.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

We did not enable a money management stop, but we did enable a % risk trailing stop for this system. Our initial protective stop for a long position is set at the 4-bar low; our initial stop for a short position is set at the 4-bar high.

In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position, and market conditions generate another short entry order.

Testing & Improving

We applied this system to a daily chart of Coca-Cola from 1970 - 1998. Sixty-five percent of the trades were profitable. The profit factor was 2.19, meaning that the system made $2.19 in profits for each dollar it lost. Figure 16 shows the resulting sample System Report.

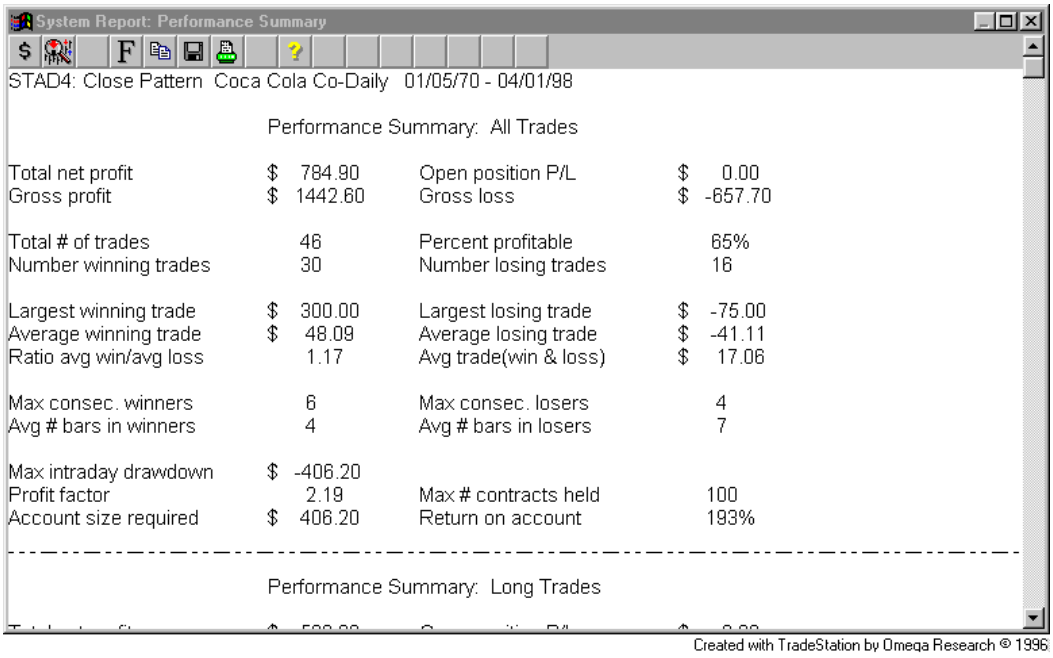


Figure 16. Sample System Report for the Consecutive Close System on a daily chart of CocaCola

Next we tested this system with the same parameters on a continuation chart of the CRB Index futures contract. Figure 17 shows the resulting chart.

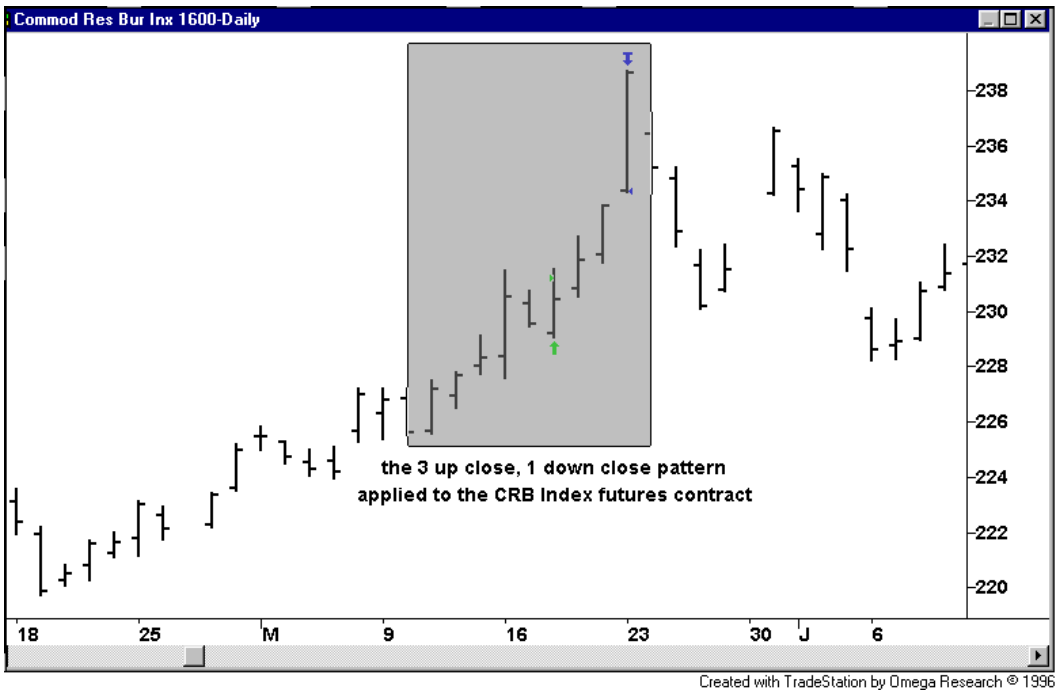


Figure 17. The Consecutive Close System applied to a CRB Index futures contract

The CRB (Commodity Research Bureau) Index, which represents a basket of 21 major commodities, is widely monitored as an early warning sign of inflation. The system earned $4,900 per contract. Sixty percent of the trades were profitable, and the system made money on both the long and short sides of the market. Figure 18 shows the resulting System Report, the Trade by Trade view.

System Report: Trade by Trade

STAD4: Close Pattern Commod Res Bur Inx 1600-Daily 06/01/93 - 07/17/98

Date	Time	Type	Cnts	Price	Signal Name	Entry P/L	Cumulative
09/07/93		Sell	1	215.75			
09/08/93		SExit	1	216.95	Money Mngmt S	$ -600.00	$ -600.00
12/14/93		Buy	1	225.85			
12/27/93		LExit	1	226.55		$ 350.00	$ -250.00
04/04/94		Sell	1	226.70			
04/06/94		SExit	1	224.40		$ 1150.00	$ 900.00
05/18/94		Buy	1	231.20			
05/23/94		LExit	1	234.35		$ 1575.00	$ 2475.00
11/18/94		Sell	1	232.55			
11/22/94		SExit	1	231.65		$ 450.00	$ 2925.00
12/28/94		Buy	1	236.40			
12/30/94		LExit	1	235.75		$ -325.00	$ 2600.00
03/02/95		Sell	1	232.65			
03/06/95		SExit	1	232.80		$ -75.00	$ 2525.00
03/29/95		Sell	1	232.30			
03/30/95		SExit	1	233.50	Money Mngmt S	$ -600.00	$ 1925.00
06/19/95		Buy	1	235.90			

Created with TradeStation by Omega Research © 1996

Figure 18. The sample System Report for the Consecutive Close System applied to the CRB Index futures contract

Suggestions for Improvement

We like this system enough to wish it generated more trades. Perhaps it would be worthwhile to test it on intraday bars, e.g. a 5-minute S&P futures chart.

Another possible improvement would be to apply a longer-term moving average (40 - 60 bars) and only take the trades in the direction of the slope of the moving average.

Momentum-Retracement System

The Momentum-Retracement System demonstrates a way of trading within an established trend. The system consists of seven components: 1) trend, 2) momentum, 3) directional movement, 4) retracements, 5) entries, 6) stops, and 7) exits.

The first step in applying the Momentum-Retracement System is to determine the trend. We use a moving-average channel and the MACD Indicator to accomplish this task.

The moving-average channel includes exponential moving averages of 15 highs, 15 lows, and 5 closes. The trend is up when the 5-bar exponential average of the closes is above 15-bar exponential average of the highs, or when the 5-bar exponential moving average of the closes is above the 15-bar exponential moving average of the highs more recently than it was below the 15-bar exponential moving average of the lows.

Conversely, the trend is down when 5-bar exponential average of the closes is below the 15-bar exponential average of the lows or when the 5-bar exponential average of the closes was below 15-bar exponential average of the lows more recently than it was above 15-bar exponential average of the highs.

The MACD Indicator must confirm the trend identified by the moving-average channel. We use a 3-10-15 MACD. This means that the MACD line represents the difference between a 3-bar and a 10-bar exponential moving average, and that the signal line of the indicator is a 15-bar exponential moving average of the MACD line. When the signal line is above zero, the trend is up, and when the signal line is below zero, the trend is down.

The second step is to evaluate the market's momentum. In this system, we use the Relative Strength Index (RSI). RSI compares the relative strength of price gains on bars that close above the previous bar's close to price losses on bars that close below the previous bar's close. A 5-bar RSI rising to 70 or higher signifies strong bullish momentum, and the RSI falling to 30 or lower indicates strong bearish momentum.

The third step is to measure the market's directional movement. In this system, we use the Directional Movement Index (DMI). DMI consists of two lines — the DMIPlus line and the DMIMinus line. These two lines represent the amount of consistent bullish "trendiness" and consistent bearish "trendiness" respectively. We construct an indicator called the DMI Spread by subtracting the DMIMinus line from the DMIPlus line. A DMI spread of +15 or higher indicates a persistent uptrend, and a DMI spread of -15 or lower indicates a persistent downtrend.

The fourth step is to identify a retracement. A retracement refers to a countertrend decline in an uptrend or a countertrend rally in a downtrend. In this system, we specify three conditions that must be met for a qualifying retracement. In an uptrend, the three conditions are: 1) Prices decline into the moving-average channel (in other words, the low of a price bar crosses below the exponential moving average of 15 highs), 2) MACD crosses below the signal line, and 3) RSI declines from above 70 to below 50 or declines by at least 30 RSI points.

In a downtrend, the three conditions are: 1) Prices rally into the moving-average channel (i.e., the high of a price bar crosses above the exponential moving average of 15 lows), 2) MACD crosses above the signal line, and 3) RSI rises from below 30 to above 50 or rises by at least 30 RSI points.

These three conditions for retracements do not have to occur on the same bar. The requirement is that all three conditions occur within 10 bars of the highest high (in an uptrend) or within 10 bars of the lowest low (in a downtrend).

The fifth step is to determine the entry price. After a qualified retracement in an uptrend, we enter a long position when prices rally one third of the way back up between the low of the retracement and the high of the uptrend. For example, if the high of the uptrend is 124, and the low of the retracement is 112, we will set our buy stop at 116 (one third of the way back up).

After a qualified retracement in a downtrend, we enter a short position when prices fall one third of the way back down between the high of the retracement and the low of the downtrend. For example, if the low of the downtrend is 112, and the high of the retracement is 124, we will set our sell stop at 120 (one third of the way back down).

The setup for a buy entry is canceled if the exponential moving average of 5 closes crosses below the exponential moving average of 15 lows or if the signal line of the MACD crosses below zero. The setup to sell short is canceled if exponential moving average of 5 closes crosses above the exponential moving average of 15 highs or if the signal line of the MACD crosses above zero.

The sixth step is to determine our initial protective stop, our breakeven stop, and our trailing stop. For a long position, the initial protective stop is set one 10-bar average true range below our entry price; for a short position, the initial stop is set a 10-bar average true range above our entry price.

We move our stop to breakeven after a closing price that gives us an open profit equal to or greater than the initial risk on the trade. For example, if we bought a stock at $80 and set our initial stop at $77, we would raise our stop to $80 (breakeven) after a close of $83 or higher.

We'll also place a profit-protect trailing stop. For a long position, we watch two trailing stops and place our profit-protect order at the higher of the two stops. The first stop is swing support to the power of two, which is defined as a price low that has at least two higher lows immediately before it and at least two higher lows immediately after it. The second stop is the exponential moving averages of 15 lows. Remember that we will place our stop at the higher of the two possibilities.

For a short position, we also watch two trailing stops and place our profit-protect order at the lower of the two stops. The first stop is swing resistance to the power of two, which can be defined as a price high that has at least two lower highs immediately before it and after it. The second stop is the exponential moving averages of 15 highs. We will place our stop at the lower of the two alternatives.

The seventh step is the exit. We exit from a long position on the next open when both the signal line of the MACD and the DMI spread fall below zero; we exit from a short position on the next

open when both the MACD signal line and the DMI spread rally above zero. Figure 19 shows the Momentum-Retracement System applied to a weekly chart of General Electric.

Figure 19. The Momentum-Retracement System applied to a weekly chart of General Electric

Our Momentum-Retracement System is fairly long and complex compared to most of the systems we have presented in the STAD Club so far. We believe, however, that the time you devote to studying this system will be time well spent.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did substantial setup work to calculate the moving average channel, momentum, and DMI. The setup, entries and exits are described next:

Setup

a) Calculate the 15-bar exponential moving averages of the highs and the lows and the 5-bar exponential moving average of the closing prices.

b) Calculate a 15-bar exponential average of the MACD (based on 3 and 10 bars).

c) Calculate the difference between the exponential average of the MACD calculated in b) and the underlying MACD.

d) Calculate the DMI spread by subtracting the DMI Minus from the DMI Plus, based on a length of 15 bars.

Long Entries

a) Determine the trend by comparing the exponential moving averages. The trend is up when: the 5-bar exponential average of the closes is above 15-bar exponential average of the highs, or when the 5-bar exponential moving average of the closes is above the 15-bar exponential moving average of the highs more recently than it was below the 15-bar exponential moving average of the lows.

b) Confirm the trend using the MACD. When the signal line is above zero, the trend is up.

c) Evaluate the momentum of the market using the RSI Indicator. A 5-bar RSI rising to 70 or higher signifies strong bullish momentum.

d) Determine the persistence of the trend using the DMI Indicator. We construct an indicator called the DMI Spread by subtracting the DMIMinus line from the DMIPlus line. A DMI spread of +15 or higher indicates a persistent uptrend.

e) Check for a retracement. In an uptrend, the three conditions are: 1) Prices decline into the moving-average channel (in other words, the low of a price bar crosses below the exponential moving average of 15 highs), 2) MACD crosses below the signal line, and 3) RSI declines from above 70 to below 50 or declines by at least 30 RSI points. These three conditions for retracements do not have to occur on the same bar. The requirement is that all three conditions occur within 10 bars of the highest high.

Short Entries

a) Determine the trend by comparing the exponential moving averages. The trend is down when: the 5-bar exponential average of the closes is below the 15-bar exponential average of the lows or when the 5-bar exponential average of the closes was below 15-bar exponential average of the lows more recently than it was above 15-bar exponential average of the highs.

b) Confirm the trend using the MACD. When the signal line is below zero, the trend is down.

c) Evaluate the momentum of the market using the RSI Indicator. A 5-bar RSI falling to 30 or lower indicates strong bearish momentum.

d) Determine the persistence of the trend using the DMI Indicator. We construct an indicator called the DMI Spread by subtracting the DMIMinus line from the DMIPlus line. A DMI spread of -15 or lower indicates a persistent downtrend.

e) Check for a retracement. In a downtrend, the three conditions are: 1) Prices rally into the moving-average channel (i.e., the high of a price bar crosses above the exponential moving average of 15 lows), 2) MACD crosses above the signal line, and 3) RSI rises from below 30 to above 50 or rises by at least 30 RSI points.

These three conditions for retracements do not have to occur on the same bar. The requirement is that all three conditions occur within 10 bars of thelowest low.

Exit Orders

a) Place protective stops. For a long position, we'll set the initial protective stop at our entry price minus the 10-bar average true range; for a short position, we'll set the initial stop at our entry price plus a 10-bar average true range.

b) We'll also place a profit-protect trailing stop:

1. For a long position, we'll watch two trailing stops and place our profit-protect order at the higher of the two stops. The first stop is swing support to the power of two, which is defined as a price low that has at least two higher lows immediately before it and at least two higher lows immediately after it. The second stop is one point below the exponential moving averages of 15 lows. We will place our stop at the higher of the two possibilities.

2. For a short position, the first stop is swing resistance to the power of two, which can be defined as a price high that has at least two lower highs immediately before it and after it. The second stop is one point above the exponential moving averages of 15 highs. We will place our stop at the lower of the two alternatives.

c) We exit from a long position on the next open when both the signal line of the MACD and the DMI spread fall below zero; we exit from a short position on the next open when both the MACD signal line and the DMI spread rally above zero.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

We created this system in modules. We wrote the criteria for the long side of the Momentum-Retracement System in one system (STAD 4: L Mom-Retrcmnt) and the criteria to follow the short side of the Momentum-Retracement System in another (STAD 4: S Mom-Retrcmnt). Then we created a third system that references the first and second system (STAD 4: Mom-Rtrcmnt). *You only need to apply the one system, STAD 4: Mom-Rtrcmnt.*

In order to reference one system from another, we used the 'IncludeSystem' EasyLanguage command. This command is very useful and commonly used. There are two main advantages to using the 'IncludeSystem' command:

a) In TradeStation 4, each system can be a maximum of 64K in size. If your system is large, it is possible to exceed this limit. One easy way to overcome this limit is to break your system into one or more systems and then combine them using the 'IncludeSystem' command. Then, each system has 64K allocated to it.

b) You can use the 'IncludeSystem' command to include instructions that you use often. For example, if there is an exit you use often, you can write it into a system and then include it any time you want to use it in a system you are writing.

Following are the instructions for the main system. Notice how short it is. Notice also that we defined inputs. You don't have to, however, if you want to be able to modify the systems you included, you have to define the inputs in the main system.

EasyLanguage Instructions: Momentum-Retracement (STAD 4: Mom-Rtrcmnt)

```
Input: XMALen(5), Len2(10);


IncludeSystem:"STAD4:S Mom-Retrcmnt",XMALEN,LEN2;
IncludeSystem:"STAD4:L Mom-Retrcmnt",XMALEN,LEN2;
```

This system merges the two systems and converts them into one, so they can be applied as one system on any given chart.

The EasyLanguage instructions for the long side of the Momentum-Retracement System are listed next. The instructions for the short side of the Momentum-Retracement System are not listed because they exactly mirror the long side instructions. The subsequent discussion also covers the long side; however, the explanations can be applied to the short side, and of course, the final main system is the two together.

EasyLanguage Instructions: Long side of the Momentum-Retracement (STAD 4: L Mom-Retrcmnt)

```
Input: XMALen(5), Len2(15);
Vars: PriceXMA(0), HighXMA(0), LowXMA(0), MACDSigLine(0), MACDDiff(0);
Vars: TrendUp(False), Momentum(False), Retracement(False), ExitPrc(0);
Vars: HighestHi(0), HighestRSI(0), LowestLo(999999), BuySwitch(False);
Vars: DMISpread(0), Fase1(False), Fase2(False);

{Calculate exponential moving averages, MACD values and DMI Spread}
PriceXMA = XAverage(Close,XMALen);
HighXMA = XAverage(High,Len2);
LowXMA = XAverage(Low, Len2);
```

```
MACDSigLine = XAverage( MACD(Close,3,10), 15);
MACDDiff = MACD(Close,3,10) - MACDSigLine;
DMISpread = DMIPlus(15) - DMIMinus(15);

{Determine trend and momentum}
TrendUp = PriceXMA > HighXMA and MACDSigLine > 0;
Momentum = RSI(Close,5) >= 70 AND DMISpread > 15;

If TrendUp and Momentum then Begin
        If Fase1 = False then Begin
                HighestHi = 0;
                HighestRSI = 0;
        End;
        Fase1 = True;
        Fase2 = False;
        If High > HighestHi then HighestHi = High;
        If RSI(Close,5) > HighestRSI thenHighestRSI = RSI(Close,5);
End;

{Check for Retracement and establish Buy Setup}
Retracement = Fase1 AND PriceXMA < HighXMA AND MACDDiff < 0 AND (RSI(Close,5)<=50 OR
        RSI(Close,5) < HighestRSI-30);

If Retracement then Begin
        If Fase2 = False thenLowestLo = 999999;
        Fase2 = True;
        Fase1 = False;
        BuySwitch = False;
End;

If Fase2 then Begin
        If Low < LowestLo then
                LowestLo = Low;
        BuySwitch = True;
End;

{Place entry order and re-set switches}
If BuySwitch and MarketPosition<> 1 then Begin
        Buy next bar at LowestLo + (HighestHi - LowestLo )/3 Stop;
End;

If MarketPosition = 1 OR (PriceXMA < LowXMA or MACDSigLine < 0) then Begin
        BuySwitch = False;
        Fase1 = False;
        Fase2 = False;
        ExitLong next bar at Market;
End;

{Set Trailing Stops and Exit orders}
If MarketPosition <> 1 then
        ExitLong next bar at Low - Average(TrueRange,10) Stop
Else
        ExitLong next bar at EntryPrice - Average(TrueRange, 10) Stop;

If SwingLow(1,Low,2,4) = 3 AND MarketPosition = 1 thenExitPrc = Low[3];

ExitPrc = MinList(LowXMA, ExitPrc);

If MarketPosition = 1 thenExitLong next bar at ExitPrc Stop;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default	Description
XMALen	5	Length, expressed in bars, used to calculate the exponential average of the closing prices.
Len2	15	Length, expressed in bars, used to calculate the exponential average of the high prices.

In addition to these inputs, we define the following variables:

```
Vars: PriceXMA(0), HighXMA(0), LowXMA(0), MACDSigLine(0), MACDDiff(0);
Vars: TrendUp(False), Momentum(False), Retracement(False), ExitPrc(0);
Vars: HighestHi(0), HighestRSI(0), LowestLo(999999), BuySwitch(False);
Vars: DMISpread(0), Fase1(False), Fase2(False);
```

Setup

We begin by calculating the exponential averages. We use the function XAverage to calculate the three averages we will be using.

```
PriceXMA = XAverage(Close,XMALen);
HighXMA = XAverage(High,Len2);
LowXMA = XAverage(Low, Len2);
```

Then, we calculate the 15-bar expoential average of the MACD line and store the resulting value in the variable MACDSigLine (we'll refer to this as the MACD signal line). We also subtract this average from the MACD line itself and store this value in MACDDiff (we'll refer to this differential as the MACD differential. It represents the nearness of the MACD line to its smoothed average). Finally, we calculate the DMI spread and store this in the variable DMISpread.

```
MACDSigLine = XAverage( MACD(Close,3,10), 15);
MACDDiff = MACD(Close,3,10) - MACDSigLine;
DMISpread = DMIPlus(15) - DMIMinus(15);
```

Once we have the values we need, we can begin our comparisons. As described, we want to check for trend, momentum, directional movement and retracement. First, we look for a trend. We check to make sure that the 5-bar average of the closing prices is greater than the 15-bar average of the high prices and that the MACD signal line is greater than zero. We store the resulting true or false value in the variable TrendUp.

Then, we check the RSI to evaluate momentum. We use the RSI function to check whether or not the RSI (using the closing prices and 5 bars) is greater than or equal to 70, and then make sure the DMI spread is greater than 15. We store the resulting value, true or false, in the variable Momentum.

```
TrendUp = PriceXMA > HighXMA and MACDSigLine > 0;
Momentum = RSI(Close,5) >= 70 AND DMISpread > 15;
```

Long Entries

Once we have performed the comparisons, we check their values. If the trend is up and there is strong bullish movement (RSI >= 70), then we do three things: First, if Fase1 is False, then we set our variables HighestHi and HighestRSI to zero. This will make sure we capture the current bar's high and RSI values in these variables. Second, we set the variable Fase1 to True and the

variable Fase2 to False. And three, we compare the high of the current bar to the value in HighestHi. If the high is greater, we store it in HighestHi. Likewise, we compare the value of the RSI to the value stored in HighestRSI and store the greater value in HighestRSI.

We use the variables Fase1 and Fase2 essentially to keep track of what we consider two distinct phases. The first phase is the up trend and the second phase is the retracement. As soon as we begin the second phase, we consider the first phase over and set Fase1 to False and set Fase2 to True, and vice versa.

```
If TrendUp and Momentum then Begin
        If Fase1 = False then Begin
                HighestHi = 0;
                HighestRSI = 0;
        End;
        Fase1 = True;
        Fase2 = False;
        If High > HighestHi then HighestHi = High;
        If RSI(Close,5) > HighestRSI thenHighestRSI = RSI(Close,5);
End;
```

Then, we check for a retracement. Four conditions must be true. First, Fase1 must be true, which means the trend is up and there is strong bullish movement, the 5-bar average of closing prices must be less then the 15-bar average of the high prices, the MACD differential must be less than zero, and one of the following must be true: the RSI is at or under 50 or the RSI is less than the HighestRSI minus 30. If they are, then the variable Retracement is set to True.

```
Retracement = Fase1 AND PriceXMA < HighXMA AND MACDDiff < 0 AND
        ( RSI(Close,5)<=50 orRSI(Close,5) < HighestRSI-30 );
```

If there is a retracement then we do three things. First, if Fase2 is False then we set the LowestLo variable to 999999 (this ensures that we will capture the low of the current bar in the variable LowestLo in the next set of instructions). Second, we set the variable Fase2 to True and set the variable Fase1 to False. Third, we set the variable BuySwitch to False.

```
If Retracement then Begin
        If Fase2 = False thenLowestLo = 999999;
        Fase2 = True;
        Fase1 = False;
        BuySwitch = False;
End;
```

If Fase2 is True, meaning that there was first an up trend and bullish market momentum and a subsequent retracement, then we perform two actions: first, we check to make sure that the current low is less than the LowestLo. If it is, then we'll store it in the variable LowestLo. And second, we set the variable BuySwitch to True.

```
If Fase2 then Begin
        If Low < LowestLo then
                LowestLo = Low;
        BuySwitch = True;
End;
```

Finally, if BuySwitch is True, which means there was a retracement, and we are currently not in a long position, then we place a buy stop order at the value stored in the variable LowestLo plus a third of the difference between the HighestHi and LowestLo. This ensures that prices have to rally a third of the way back up between the low of the retracement and the high of the uptrend.

```
If BuySwitch and MarketPosition<> 1 then Begin
        Buy next bar at LowestLo + (HighestHi - LowestLo )/3 Stop;
End;
```

Once we are in a long position, or if 5-bar average of the closing prices is less than the 15-bar average of the low prices or the MACD signal line is less than zero, we will re-set our variables, BuySwitch, Fase1 and Fase2 to False.

```
If MarketPosition = 1 or (PriceXMA < LowXMA OR MACDSigLine < 0) then Begin
        BuySwitch = False;
        Fase1 = False;
        Fase2 = False;
        ExitLong next bar at Market;
End;
```

Long Exits

The next set of instructions places exit orders. First, we check to see if we are in a long position. If we are not, then we place the exit order at the low of the current bar minus the 10-bar average of the true range. If we are, we place the exit order at the entry price minus the 10-bar average of the true range.

Consider this set of instructions. When we are not in a long position, we place an order to exit a long position. This statement enables us to place an exit for the bar of entry. As soon as we are in a long position, this exit order takes effect.

```
If MarketPosition <> 1 then
        ExitLong next bar at Low - Average(TrueRange,10) Stop
Else
        ExitLong next bar at EntryPrice - Average(TrueRange, 10) Stop;
```

We also wanted to place a profit-protecting trailing stop. We consider two prices, the low of the swing resistance and the exponential moving averages of the low prices. First we determine when a swing low has occurred and we are in a long position, we use the functions SwingLow and MarketPosition, respectively, to do this. When these two conditions are true, we store the low of the swing bar in the variable ExitPrc. We then use the MinList function to determine which value is lower, the 15-bar exponential moving average of the low prices (LowXMA) or the value stored in ExitPrc. Whichever is lower is stored in the variable ExitPrc. Then, we place a stop order to exit our long position at the price stored in the variable ExitPrc.

```
If SwingLow(1,Low,2,4) = 3 AND MarketPosition = 1 thenExitPrc = Low[3];
ExitPrc = MinList(LowXMA, ExitPrc);
If MarketPosition = 1 thenExitLong next bar at ExitPrc Stop;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified $10,000 as the trade size and 100 shares as the lot size.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

We did not enable a money management stop or % risk trailing stop for this system. Our initial protective stop for a long position is set one ten-bar average true range below our entry price; for a short position, the initial stop is set one ten-bar average true range above our entry price. Our stop is moved to breakeven after a closing price that gives us an open profit equal to or greater than the initial risk on the trade.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point*

value, you would type the maximum number of dollars you are willing to risk per contract traded.

In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position, and market conditions generate another short entry order.

Testing & Improving

We applied the Momentum Retracement System to a weekly chart of General Electric. The total net profit was $3,032 per 100 shares. Sixty-seven percent of the trades were profitable with a ratio of average win to average loss of 2.85. The system had seven consecutive winning trades but only three consecutive losing trades. Momentum Retracement let profits run by staying in winning trades for an average of 20 bars but cut losses short by exiting losing trades in only three bars. The profit factor was excellent at 5.70 – our system made $5.70 in profits for each dollar it lost. The maximum drawdown was only $787.50 per 100 shares during the test period of more than 20 years. The system was profitable on both the long and short sides. Figure 20 shows the sample System Report for the weekly chart of General Electric.

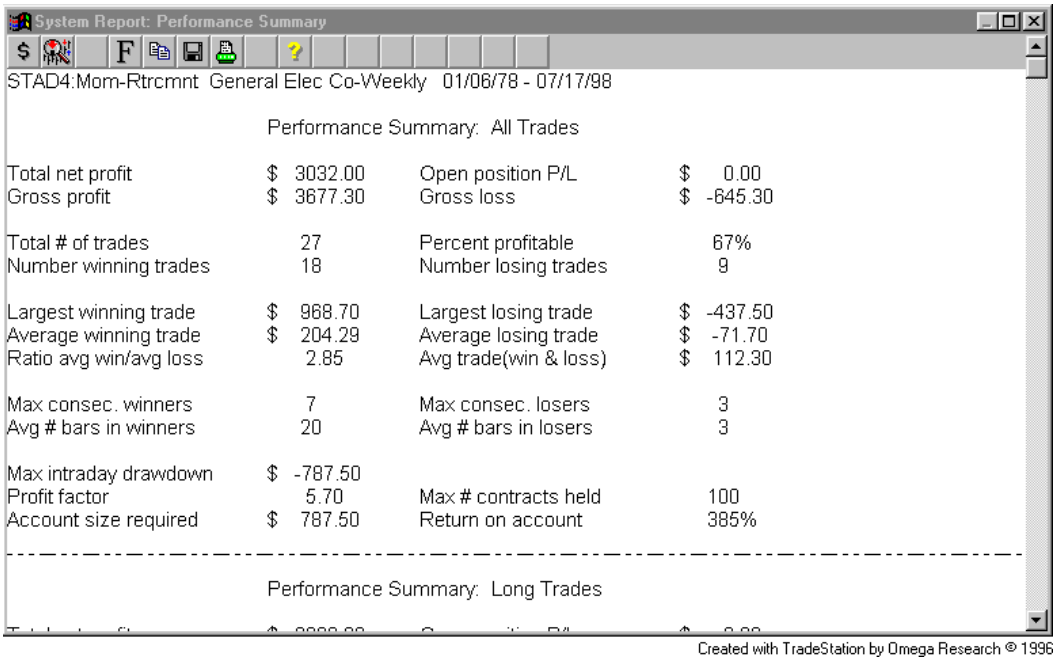


Figure 20. Sample System Report for the Momentum-Retracement System on a weekly GE chart

Suggestions for Improvement

This system was tested without any optimization. The results could almost certainly be improved by proper optimization. Inputs that could be optimized include the values for the EMA of highs, EMA of lows, EMA of closes, MACD, RSI, and DMI spread. We could also optimize the levels of the RSI and DMI spread (currently 70 – 30 for RSI and +15, -15 for the DMI spread). The percent of the countertrend move that must be "taken back" by a resumption of the major trend is another candidate for optimization (the percent is currently fixed at 33%).

The system's stops could also be optimized. The initial protective stop (currently set at one average true range from our entry point) could be optimized for the best value in a range of 1/2 of an ATR to 3 ATRs.

The breakeven stop for this system has a floor equal to or greater than the initial risk. For example, if the risk on a trade was $500, we would move our stop to breakeven when our open

profits were equal to (or greater than) $500. The breakeven stop could be optimized for values between 75% of the initial risk and two times the initial risk.

Three Exponential Moving Average Crossover System

In Chapter 1, General System Development Concepts, we used a triple moving average system to illustrate the basics of system optimization. We applied one of TradeStation's built-in systems, the MovAvg(3) Crossover system to IBM daily data. Then we optimized the length of the three moving averages to fine-tune a system that was already producing profitable results. We found that the 4-10-14 combination of simple moving averages performed much better than the default values of 4-9-18 on the market we were studying.

Now, we'd like to see if we can improve the system further. We'll start by changing from simple moving averages to exponential moving averages. We don't think that this will make too much of a difference because simple moving averages and exponential moving averages are fairly similar.

A simple moving average adds the data (generally closing prices) for a specified number of bars and divides the total by the number of bars in the lookback period. For example, a 10-bar simple moving average of closes adds the closing prices of the past 10 bars and divides the total by 10. The term *moving* is included in the name of the indicator because the average moves forward with each new bar that is created. In other words, the 10-bar simple moving average always looks back to just the most recent 10 bars. Of course, you can construct simple moving averages of any number of bars. The smaller the number of bars, the more sensitive the simple moving average. For example, a 5-bar SMA is more sensitive than a 40-bar SMA.

Although the simple moving average is a good indicator, we have a slight preference for another indicator — the exponential moving average. The EMA addresses two shortcomings of the SMA. In the SMA, each data point counts exactly the same as any other data point in the lookback period, and the SMA ignores all data points that occurred before the lookback period. For example, in a 50-bar SMA of closing prices, each of the 50 closes counts the same as any other close, and the 50-bar SMA ignores data earlier than 50 bars ago.

Many traders believe that more recent closing prices should be given more statistical weight, and that the moving average should be calculated in such a way that it takes into consideration the data that occurred before the lookback period.

Therefore, the EMA assigns more weight to recent data and does include all the available data points in its calculation. Through geometric progression, each older price is assigned less weight as the EMA moves forward in time. Thus, the effects of the earliest prices in the data series are never eliminated, but they do diminish as more prices are included in the calculation.

In the MovAvg(3) Crossover System that we tested earlier in this volume, the basic rules were as follows:

7. Buy when SMA1 is greater than SMA2, and SMA2 is greater than SMA3.
8. Sell when SMA1 is less than SMA2, and SMA2 is less than SMA3.

We noticed that this system can be very slow to exit from a position that has large open profits. When a market has moved strongly in the direction of our trade, the three moving averages can pull very far apart. Not exiting our trade until the conditions are met to actually reverse our position (from long to short or short to long) can result in the loss of an unacceptable amount of our open profit.

In an attempt to solve this problem, we wrote a revised set of rules for our Three Exponential Moving Averages Crossover System. In our new 3 XAvg Crossover System, we'll call the fastest moving average Avg1, the second fastest moving average Avg2, and the third fastest moving average Avg3. The new rules are as follows:

1. Buy when Avg1 crosses over Avg2 AND when Avg2 is greater than Avg3.
2. Sell when Avg1 crosses under Avg2 and Avg2 is less than Avg3.

We'll exit our long positions on the open of the next bar when Avg1 is less than Avg2. Likewise, we'll exit our short positions on the open of the next bar when Avg1 is greater than Avg2.

We'll also place a trailing stop for the first bar, exiting our long position at the low price minus the 4-bar average of the range. On subsequent bars, the stop price will be the initial stop price plus the value that results from subtracting the stop price from the low and dividing by 4. For the short side, on the first bar, we'll exit our position at the high price plus the 4-bar average of the range. On subsequent bars, the stop price will be the initial stop price minus the value that results from subtracting the high from the initial stop price and dividing by 4.

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. The long and short entries reverse your position, whereas the exits close out your existing position and exit you from the market. We also performed some setup work, which involved identifying the Inside Day and subsequent breakouts. The setup, entry and exits are described next.

Setup

a) Calculate the three exponential moving averages.

Long Entries

a) When Avg1 crosses over Avg2 and when Avg2 is greater than Avg3, buy on the next bar at the open.

Short Entries

a) When Avg1 crosses under Avg2 and Avg2 is less than Avg3, sell on the next bar at the open.

Exits

a) Exit all long positions on the next bar at the open when Avg1 is less than Avg2.

b) Exit all short positions on the next bar at the open when Avg1 is greater than Avg2.

c) On the first bar of a long position, place a trailing stop that exits at the low minus the 4-bar average of the range. On the first bar of a short position, place a trailing stop that exits at the high plust the 4-bar average of the range.

d) On subsequent bars, place a trailing stop to exit long positions at the original stop price plus the value that results from subtracting the stop price from the low and dividing by 4. We'll place a trailing stop to exit short positions at the initial stop price minus the value that results from subtracting the high from the initial stop price and dividing by 4.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Three Exponential Moving Averages (STAD 4: 3XAverages)

```
Inputs: Price(Close), AvgLen1(6), AvgLen2(12), AvgLen3(28);
Vars: Avg1(0), Avg2(0), Avg3(0), MP(0), StopPrice(0);

Avg1 = XAverage(Price, AvgLen1);
Avg2 = XAverage(Price, AvgLen2);
Avg3 = XAverage(Price, AvgLen3);
```

```
If Avg1 Crosses Over Avg2 AND Avg2 > Avg3 then Buy next bar at Open;
If Avg1 Crosses Under Avg2 AND Avg2 < Avg3 then Sell next bar at Open;

If Avg1 < Avg2 thenExitLong next bar at Open;
If Avg1 > Avg2 thenExitShort next bar at Open;

MP = MarketPosition;

If MP = 1 and MP[1] <> 1 thenStopPrice = Low - Average(Range,4);

If MP = 1 then Begin
        ExitLong ("ExitTrade") next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/4;
End;

If MP = -1 and MP[1] <> -1 then StopPrice = High + Average(Range, 4);

If MP = -1 then Begin
        ExitShort (“ExitTrade”) next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

IMPORTANT NOTE: *The instructions above in the bold font are the trailing stop instructions for the short side of the system. They were inadvertently left out of the system on your CD. For this system to work properly, you will have to open the system in the PowerEditor and type these instructions. Once you verify the system, you can apply it to your chart.*

Inputs

Following is the list of the inputs we used in this system:

Inputs	Default	Description
Price	Close	Price on which the exponential moving averages will be based.
AvgLen1	6	Length, expressed in bars, for the fast exponential moving average.
AvgLen2	12	Length, expressed in bars, for the medium exponential moving average.
AvgLen3	28	Length, expressed in bars, for the slow exponential moving average.

In addition to these inputs, we identified the following variables:

```
Vars: Avg1(0), Avg2(0), Avg3(0), MP(0), StopPrice(0);
```

SetUp

The exponential moving average values are calculated and assigned to variables. This allows for easy reference later, without recalculation of the XAverage function.

```
Avg1 = XAverage(Price, AvgLen1);
Avg2 = XAverage(Price, AvgLen2);
Avg3 = XAverage(Price, AvgLen3);
```

Long Entries

A buy order will be triggered on the open of the next bar if Avg1 (fast average) crosses above Avg2 (medium average) and Avg2 is greater than Avg3 (slow average).

```
If Avg1 crosses over Avg2 AND Avg2 > Avg3 then Buy next bar at Open;
```

Short Entries

A sell order will be triggered on the open of the next bar if Avg1 crosses below Avg2 and Avg2 is less than Avg3.

```
If Avg1 crosses under Avg2 AND Avg2 < Avg3 then Sell next bar at Open;
```

Exit Orders

If Avg1 is greater than Avg2, nearly the opposite of the first part of the Buy Entry criteria, a long exit will be triggered on the open of the next bar.

```
If Avg1 < Avg2 then ExitLong next bar at Open;
```

If Avg1 is less than Avg2, nearly the opposite of the first part of the sell entry criteria, a short exit will be triggered on the open of the next bar.

```
If Avg1 > Avg2 then ExitShort next bar at Open;
```

The value of the MarketPosition function is assigned to the variable MP so that we can reference previous market position values.

```
MP = MarketPosition;
```

If the current market position is long (1) and the market position on the previous bar was short or flat (-1 or 0), a stop price is calculated by subtracting the average range of four bars from the low. Thus, the stop price is calculated each time the market position becomes long.

```
If MP = 1 and MP[1] <> 1 then StopPrice = Low - Average(Range,4);
```

If the current market position is long (1) an long exit order is generated on the stop price (StopPrice) calculated above. In addition, this trailing stop is moved up by ¼ of the StopPrice value.

```
If MP = 1 then Begin
        Exitlong ("ExitTrade") next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/4;
End;
```

If the current market position is short (-1) and the market position on the previous bar was long or flat (1 or 0), a stop price is calculated by adding the average range of four bars to the high. Thus, the stop price is calculated each time the market position becomes short.

```
If MP = -1 and MP[1] <> -1 then StopPrice = High + Average(Range, 4);
```

If the current market position is short (-1) an long exit order is generated on the stop price (StopPrice) calculated above. In addition, this trailing stop is moved up by ¼ of the StopPrice value.

```
If MP = -1 then Begin
        ExitShort (“ExitTrade”) next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified 1 as the default contract size to trade.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or a dollar amount per contract/share you want to risk before exiting from the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out.When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

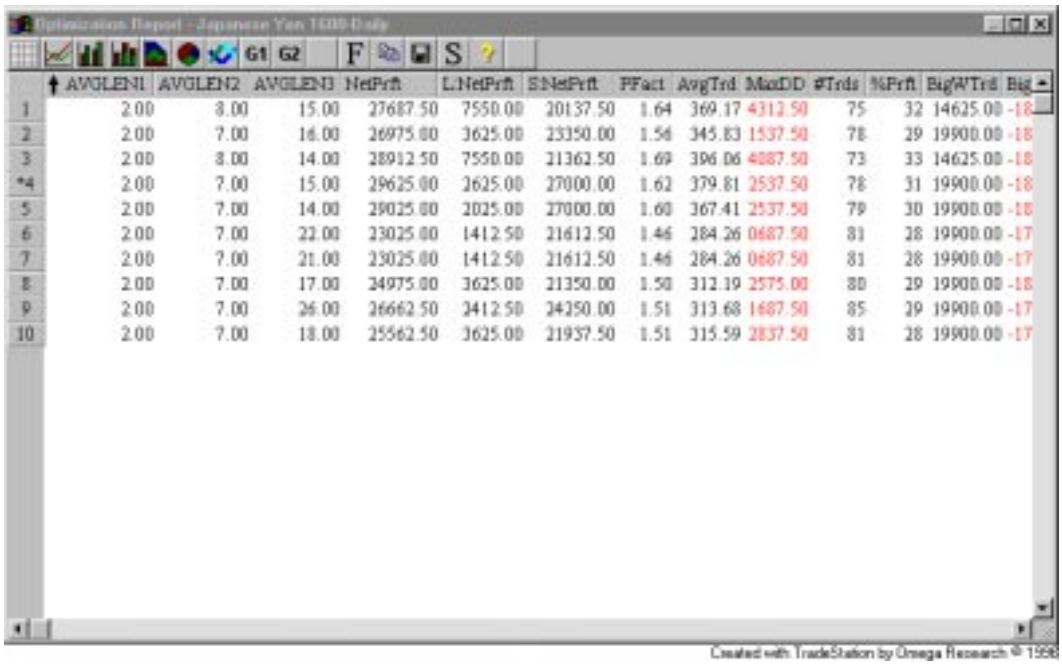
In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position, and market conditions generate another short entry order.

Testing & Improving

We optimized the length of the three exponential moving averages to determine the combination that produced the best results for our 3 XAvg Crossover System. The values we tested were 2 through 6 for Avg1, 7 through 11 for Avg2, and 14 through 20 for Avg3. We also enabled a $1,000 money management stop. Figure 26 shows the resulting system applied to a daily chart of the Yen.

Figure 26. The Three Exponential Moving Averages Crossover System applied to a daily chart of the Yen

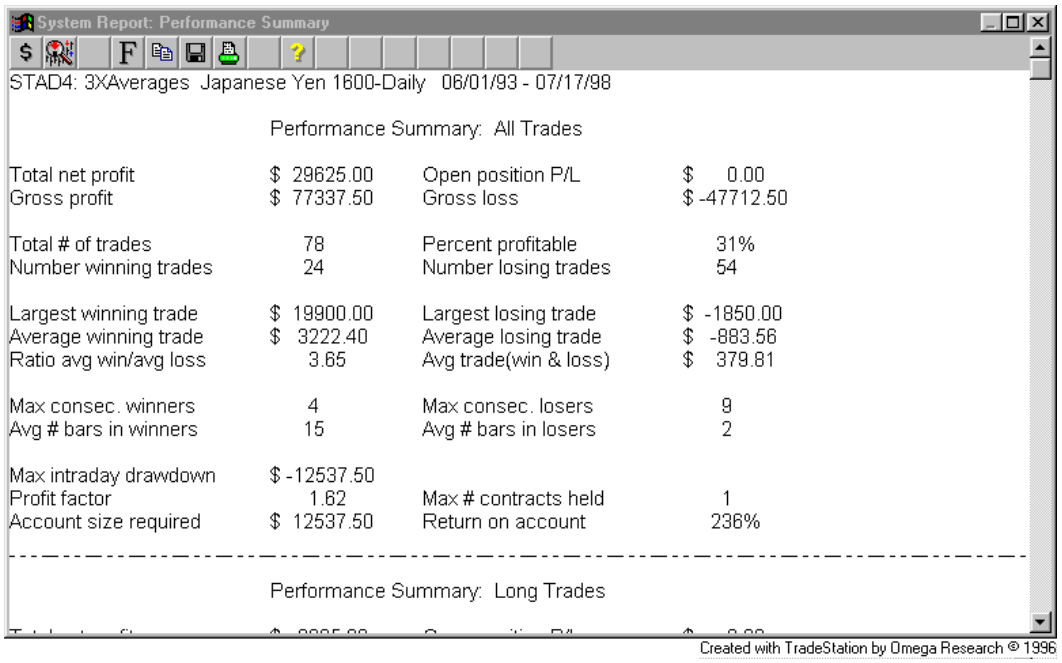
When we performed the test on a daily continuation chart of Japanese Yen futures, the strongest combination of moving averages was 2, 7, 15. Figure 27 shows the supporting Optimization Report.



	AVGLEN1	AVGLEN2	AVGLEN3	NetPrft	L:NetPrft	S:NetPrft	PFact	AvgTrd	MaxDD	#Trds	%Prft	BigWTrd	Big
1	2.00	8.00	15.00	27687.50	7550.00	20137.50	1.64	369.17	4312.50	75	32	14625.00	-18
2	2.00	7.00	16.00	26975.00	3625.00	23350.00	1.56	345.83	1537.50	78	29	19900.00	-18
3	2.00	8.00	14.00	28912.50	7550.00	21362.50	1.69	396.06	4887.50	73	33	14625.00	-18
*4	2.00	7.00	15.00	29625.00	2625.00	27000.00	1.62	379.81	2537.50	78	31	19900.00	-18
5	2.00	7.00	14.00	29025.00	2025.00	27000.00	1.60	367.41	2537.50	79	30	19900.00	-18
6	2.00	7.00	22.00	23025.00	1412.50	21612.50	1.46	284.26	0687.50	81	28	19900.00	-17
7	2.00	7.00	21.00	23025.00	1412.50	21612.50	1.46	284.26	0687.50	81	28	19900.00	-17
8	2.00	7.00	17.00	24975.00	3625.00	21350.00	1.50	312.19	2575.00	80	29	19900.00	-18
9	2.00	7.00	26.00	26662.50	3412.50	24250.00	1.51	313.68	1687.50	85	29	19900.00	-17
10	2.00	7.00	18.00	25562.50	3625.00	21937.50	1.51	315.59	2837.50	81	28	19900.00	-17

Figure 27. The Optimization Report for the Three Exponential Averages System

The system generated a total net profit of $29,625 per contract during the approximately 5-year period. Figure 28 shows the sample System Report.



System Report: Performance Summary

STAD4: 3XAverages Japanese Yen 1600-Daily 06/01/93 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 29625.00	Open position P/L	$ 0.00
Gross profit	$ 77337.50	Gross loss	$ -47712.50
Total # of trades	78	Percent profitable	31%
Number winning trades	24	Number losing trades	54
Largest winning trade	$ 19900.00	Largest losing trade	$ -1850.00
Average winning trade	$ 3222.40	Average losing trade	$ -883.56
Ratio avg win/avg loss	3.65	Avg trade(win & loss)	$ 379.81
Max consec. winners	4	Max consec. losers	9
Avg # bars in winners	15	Avg # bars in losers	2
Max intraday drawdown	$ -12537.50		
Profit factor	1.62	Max # contracts held	1
Account size required	$ 12537.50	Return on account	236%

Performance Summary: Long Trades

Created with TradeStation by Omega Research © 1996

Figure 28. The sample System Report for the Three Exponential Averages System applied to a daily chart of the Yen

The trades were 31% profitable with a ratio of average win to average loss of 3.65. The average trade (win + loss) was $379.81. Even if we deduct $50 - $100 per trade for slippage and

commission, the system still looks promising. Other positive factors include that the system was profitable on both the long and short sides, and that it stayed in the average winning trade more than seven times as long as the average losing trade.

Suggestions for Improvement

The most negative feature of this system's performance was its dependence on the largest winning trade for the good results. This, of course, is not unusual in a trend-following system, as the whole purpose of this type of system is to capture large profits in a big trend and to minimize losses in periods of choppy market activity. Although having the largest winning trade represent a major portion of the system's profits is a potential problem (what if the market doesn't provide an exceptional trend again before our money or patience runs out?), the solution is clearly not to reduce the size of the largest winning trade. The most practical solution is found in portfolio management. Since no one really knows for sure when and where the next explosive trend will occur, most successful trend-followers diversify their portfolios to increase their chances of participation in an exceptional trend. Trading a carefully selected "basket" of stocks and/or commodities rather than limiting your efforts to only one market has the potential to smooth your equity curve and increase your profits.

Investing a Fixed Dollar Amount

A somewhat overlooked factor when back-testing a system on stocks (or any instrument whose price changes significantly over time) is that a 100-share trade on a 10 dollar stock is not the same as a 100-share trade on a 100 dollar stock. Because the dollar amount required to make these transactions is so different, the return of these two trades cannot be compared in dollar amounts (a return of $1,000 for a $5,000 investment is very good whereas the same return on $100,000, while still good, is not nearly as significant).

One way to avoid this is to invest a fixed dollar amount instead of a fixed number of shares. Therefore, we created the NumUnits function to perform this calculation for you.

The function is listed below, and it is also included on your STAD Club Volume 4 CD.

EasyLanguage Instructions: NumUnits Function (NumUnits)

```
Inputs: Amnt(numeric), MinLot(numeric);
NumUnits = IntPortion( IntPortion(Amnt / (Close*BigPointValue) ) / MinLot ) * MinLot;
```

Let's see how we developed this function. First, we multiply the closing price by the value returned by the BigPointValue function. The BigPointValue function returns the dollar value of a 1 point move for the specific stock). Then, we divide the amount of dollars we want to invest (specified using the Amnt input) by the resulting value. For example, if we decide to invest $15,500 in lots of 100 shares, and the last price of the stock is $65 a share, the first operation will be:

```
15,500 / ( 65 * 1 )
```

The result is 238.46 shares. Now, we can't trade fractions, so we will truncate the decimals of this expression using the IntPortion function (built in to TradeStation). The result is:

```
IntPortion( 15,000 / (65 * 1 )
```

The resulting number of shares is 238. However, we only want to trade in lots of 100 shares, so we divide our 238 shares by our minimum lot amount (specified using the MinLot input), truncate the decimals once more, and then multiply the result by the minimum lot:

```
IntPortion( IntPortion(15500 / (65 * 1) ) / 100 ) * 100
```

The final result is 200. Therefore, we would be able to trade 200 shares of a $65 stock with $15,500.

CHAPTER 3

Support & Resistance Systems

Support & Resistance systems are designed for sideways or directionless markets, and they typically have the following attributes:

- They buy low and sell high in an attempt to take advantage of the sideways price movement characterizing directionless markets.
- They have a high number of winning trades, with small profits on each trade. They sell as the market goes higher and take small losses until the market finally turns down and results in a profitable trade.
- They are easier to trade emotionally.

By design, these systems miss the big move — they usually have small profits and larger losses as markets trend. The system keeps shorting a market that is in an uptrend or buying a market that is in a downtrend. Therefore, when traders use Support &Resistance systems, they use them within a group of systems that also includes trending systems and perhaps one or more volatility systems.

In this chapter, we present two Support & Resistance systems, designed to make the most of the sideways movement of a market. We also show how these two systems can be used in conjunction to more effectively take advantage of market movement.

In This Chapter

Spread System: Beta Up

Market Guide Beta is traditionally defined as a measure of a stock's price volatility relative to the S&P 500 Index. Up Beta is the volatility of the company's stock relative to the index for the period in which the index increases, and Down Beta is the volatility of a stock relative to the index for the periods in which the index decreases.

Market Guide Beta returns the slope of the 60-month regression line of the percentage price change of the stock relative to the percentage change of the S&P 500, adjusted for regression tendencies reported by Blume.

A stock with a strong Up Beta (greater than 1) will tend to move upwards — accompanying and usually exceeding moves of the S&P 500 Index — so we can design a system around this concept in an effort to capture the reactions of the stock to the market. If the S&P 500 Index moves upward, we would expect the stock to rally with it. If the index makes the move and the stock hasn't reacted, we may want to establish a long position in an effort to capture the rally the stock likely will experience as it catches up to (or even exceeds) the S&P's performance.

Inversely, a stock with a strong Beta Down (greater than 1) will tend to decline — accompanying and usually exceeding an S&P 500 Index decline. Therefore, if a stock with a strong Down Beta is over-performing the S&P 500 Index, and the index drops without the stock accompanying it immediately, we would expect the stock to decline and we'll look to establish a short position.

We will design two systems with this idea. Once for strong Beta Up stocks, the second for strong Beta Down stocks. This section describes the Beta Up system, the next section describes the Beta Down system.

Both systems will calculate the percent change of the traded stock and of the S&P 500 Index of the last 10 days. The Beta Up system will then look for a day in which the S&P's percent change rose while the stock remained unchanged or declined. When this happens, we will buy on the next bar at the high of the current day.

To exit, will use the trailing stop we have used in previous volumes of the STAD Club. This consists of obtaining the low of the bar of entry and subtracting from it the average range of the last four bars. This will be our exit point for the first bar after establishing a long position. From the second bar, we will calculate the distance from the low to our previous exit price, and divide it by three. We will add this value to our previous exit price obtain our exit for the next day.

Defining your Trading Rules

In this system, we defined only long entries and exit orders. We also did some setup work, which consists of calculating the percent change between the two data series. The setup, entry and exits are described next.

Setup

a) Divide the close of the current bar by the close 10 bars ago. Make sure to check for division by zero before dividing.

b) Divide the close of current bar for the second data series by the close of the second series 10 bars ago. Again, make sure to check for division by zero before dividing.

Long Entries

a) When we are not in a long position and the percent change for the first data series is less than the percent change for the second data series, and the 8-bar exponential average of the closing prices is greater than the 15-bar exponential average of the closing prices, then we will buy on the next bar at the open.

Exits

a) On the first bar of entry, we will set a trailing stop at the low of the current bar minus the 4-bar average of the range.

b) On subsequent bars, our exit price will be the initial stop price plus the value resulting from subtracting the initial stop price from the low and dividing by 3.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system. This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Spread System: Beta Up (STAD 4: Spread B+)

```
Inputs: Price1(Close), Price2(Close of Data2), Len(10), Trade(15000), Lot(100);
Vars: PcntChange1(0), PcntChange2(0);
Vars: MP(0), StopPrice(0);

If Price1[Len] <> 0 then PcntChange1 = Price1 / Price1[Len];
If Price2[Len] <> 0 thenPcntChange2 = Price2 / Price2[Len];

If MarketPosition <> 1 and PcntChange1 < PcntChange2 And XAverage(Close,8) > XAverage(C,15) then
        Buy NumUnits(Trade, Lot) shares next bar at Open;

MP = MarketPosition;
if MP = 1 and MP[1] <> 1 thenStopPrice = Low - Average(Range,4);
If MP = 1 then Begin
        ExitLong ("ExitLTrade") next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

Inputs

Following is the list of the inputs we used in this system:

Inputs	Default	Description
Price1	Close	Data used to calculate the first element of the spread.
Price2	Close of Data2	Data used to calculate second element of the spread.
Len	10	Period, expressed in bars, used to calculate the percent change of Price1 and Price2.
Trade	15,000	Investment, in dollars, per trade.
Lot	100	Miminum lot size per transaction.

In addition to these inputs, we identified the following variables:

```
Vars: PcntChange1(0), PcntChange2(0);
Vars: MP(0), StopPrice(0);
```

Setup

First, we calculate the percent change for both data series, we divide the close of the current bar with the close of 10 bars ago (for both data series). Since we will be dividing, we first make sure that our divisor, Price1 and Price2 of Len bars ago, are not equal to zero. We store the resulting percent change values in the variables PcntChange1 and PcntChange2.

```
If Price1[Len] <> 0 thenPcntChange1 = Price1 / Price1[Len];
If Price2[Len] <> 0 thenPcntChange2 = Price2 / Price2[Len];
```

Long Entries

If this system is not in a long position and the percent change of the first data stream is less than the percent change of the second data stream, and the 8-bar exponential average of the closing prices is greater than the same 15-bar average, then we will look to establish a long position at the next bar's open.

We specify the number of shares to buy using the NumUnits function. Please refer to the section at the end of Chapter 2, titled, "Investing a Fixed Dollar Amount," for a description of the NumUnits function and how it calculates the number of shares to buy.

```
If MarketPosition <> 1 and PcntChange1 < PcntChange2 and XAverage(Close,8) > XAverage(C,15) then
        Buy NumUnits(Trade, Lot) shares next bar at Open;
```

Exit Orders

We will use a version of the trailing stops that we have used in previous volumes of the STAD Club. We will use the variable MP to store the value of the keyword MarketPosition. This with the intention of being able to reference the values of MarketPosition of previous bars.

```
MP = MarketPosition;
```

If we are in a long position, and one bar ago we were not in a long position, then we will assign the low of the current bar minus the 4-bar average of the range to the variable StopPrice.

```
If MP = 1 and MP[1] <> 1 thenStopPrice = Low - Average(Range,4);
```

If we are in a long position, we will place a stop order to exit from the position at the StopPrice or anything lower. Then, we will calculate a third of the distance between the low of the current bar and the StopPrice and we will add that to the current stop price; the resulting value is our new StopPrice.

```
If MP = 1 then Begin
        Exitlong next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

General System Format

When we apply this system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified 1 as the default contract size to trade.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or a dollar amount per contract/share you want to risk before exiting from the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited.When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored.

Testing & Improving

Please see the description of the next system, Beta Down, for a description of the system results and suggestions for improvement. The two systems, although, they are two separate systems, were designed to be used together, so their use is described in the next section.

Spread System: Beta Down

Market Guide Beta is traditionally defined as a measure of a stock's price volatility relative to the S&P 500 Index. Up Beta is the volatility of the company's stock relative to the index for the period in which the index increases, and Down Beta is the volatility of a stock relative to the index for the periods in which the index decreases.

Market Guide Beta returns the slope of the 60-month regression line of the percentage price change of the stock relative to the percentage change of the S&P 500, adjusted for regression tendencies reported by Blume.

A stock with a strong Up Beta (greater than 1) will tend to move upwards — accompanying and usually exceeding moves of the S&P 500 Index — so we can design a system around this concept in an effort to capture the reactions of the stock to the market. If the S&P 500 Index moves upward, we would expect the stock to rally with it. If the index makes the move and the stock hasn't reacted, we may want to establish a long position in an effort to capture the rally the stock likely will experience as it catches up to (or even exceeds) the S&P's performance.

Inversely, a stock with a strong Beta Down (greater than 1) will tend to decline — accompanying and usually exceeding an S&P 500 Index decline. Therefore, if a stock with a strong Down Beta is over-performing the S&P 500 Index, and the index drops without the stock accompanying it immediately, we would expect the stock to decline and we'll look to establish a short position.

We will design two systems with this idea; one for strong Beta Up stocks, the second for strong Beta Down stocks. This section describes the Beta Down system, and the previous section describes the Beta Up system.

Both systems will calculate the percent change of the traded stock and of the S&P 500 Index of the last 10 days. The Beta Down system will look for a day in which the S&P 500 index percent change declines and where the stock's remains at zero or increases. When a day such as this is found, it will establish a short entry order (sell to open) at the low of the current bar.

To exit, we'll use the trailing stop we have used in previous volumes of the STAD Club. This consists of obtaining the high of the bar of entry and adding to it the average range of the last four bars. This will be our exit point for the first bar after establishing a short position. From the second bar, we will calculate the distance from our initial stop price to the high of the current bar and divide it by three. We will subtract this value from the previous stop price in order to obtain our exit for the next day.

Figure 1 shows the Beta Down applied to a daily chart of Arcadia Financial and the S&P 500 Index:

Figure 1. Beta Down System applied to Arcadia Financial and the S&P 500 Index

Defining your Trading Rules

In this system, we defined only short entries and exit orders. We also did some setup, which involved calculating the percent change for the two data series. The setup, entry and exits are described next.

Setup

a) Divide the close of the current bar by the close 10 bars ago. Make sure to check for division by zero before dividing.

b) Divide the close of current bar for the second data series by the close of the second series 10 bars ago. Again, make sure to check for division by zero before dividing.

Short Entries

a) When we are not in a short position and the percent change for the second data series is less than the percent change for the second data series one bar ago, and the percent change for the first data series is greater than or equal to the percent change for the first data series one bar ago, then we will place a stop order for the next bar at the low of the current bar.

Exits

a) On the first bar of entry, we will set a trailing stop at the high of the current bar plus the 4-bar average of the range.

b) On subsequent bars, our exit price will be the initial stop price minus the value resulting from subtracting the high from the initial stop price and dividing by 3.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Spread System: Beta Down (STAD 4: Spread B-)

```
Inputs: Price1(Close), Price2(Close of Data2), Len(10), Trade(15000), Lot(100);
Vars: PcntChange1(0), PcntChange2(0);
Vars: MP(0), StopPrice(0);

If Price1[Len] <> 0 thenPcntChange1 = Price1 / Price1[Len];
If Price2[Len] <> 0 thenPcntChange2 = Price2 / Price2[Len];

If MarketPosition <> -1 and PcntChange2 < PcntChange2[1] and PcntChange1 >= PcntChange1[1] then
        Sell NumUnits(Trade,Lot) shares next bar at Low Stop;
MP = MarketPosition;
if MP = -1 and MP[1] <> -1 thenStopPrice = High + Average(range,4);
If MP = -1 then Begin
        ExitShort ("ExitSTrade") next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

IMPORTANT NOTE: *When you compare the above instructions in bold to the same line in the system in the PowerEditor, you will notice a difference. The above instructions are correct; please correct your system in the PowerEditor.*

Inputs

Following is the list of the inputs we used in this system:

Inputs	Default	Description
Price1	Close	Data used to calculate the first element of the spread.
Price2	Close of Data2	Data used to calculate second element of the spread (we used the S&P 500 Index).
Len	10	Period, expressed in bars, used to calculate the percent change of Price1 and Price2.
Trade	15,000	Investment, in dollars, per trade.
Lot	100	Minimum lot size per transaction.

In addition to these inputs, we identified the following variables:

```
Vars: PcntChange1(0), PcntChange2(0);
Vars: MP(0), StopPrice(0);
```

If Price1 and Price2 of Len bars ago is something other than zero, we will divide the current value with the value of Len bars ago. These IF-THEN statements are included in order to avoid a division by zero.

```
If Price1[Len] <> 0 thenPcntChange1 = Price1 / Price1[Len];
If Price2[Len] <> 0 thenPcntChange2 = Price2 / Price2[Len];
```

Short Entries

If this system is not in a short position and the percent change of the second data stream is less than the percent change from the second data stream one bar ago, and the percent change from the first data series is greater than or equal to the same value one bar ago, then we will look to establish a short position at the low of the current bar or anything lower.

We use the NumUnits function to determine the number of shares to sell. Please see the last section in Chapter 2, titled, "Investing a Fixed Dollar Amount," for a description of how the function calculates the number of shares to trade.

```
If MarketPosition <> -1 and PcntChange2 < PcntChange2[1] and PcntChange1 >= PcntChange1
        then Sell NumUnits(Trade,Lot) shares next bar at Low stop;
```

Exit Orders

We will use a version of the trailing stops that we have used in previous volumes of the STAD Club. We will use the variable MP to store the value of the keyword MarketPosition. This enables us to be able to reference the values of MarketPosition of previous bars.

```
MP = MarketPosition;
```

If we are in a short position, and one bar ago we were not in a short position, then we will assign the high of the current bar plus the four bar average of the range to the variable StopPrice.

```
If MP = -1 and MP[1] <> -1 thenStopPrice = High + Average(Range,4);
```

If we are in a short position, we will place a stop order to exit from the position at the StopPrice or anything higher. Then we will calculate a third of the distance between the StopPrice and the high of the current bar and we will subtract this value from the current stop price; the resulting value is our new StopPrice.

```
If MP = -1 then Begin
        Exitshort next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified 1 as the default contract size to trade.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or a dollar amount per contract/share you want to risk before exiting from the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited.When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position, and market conditions generate another short entry order.

Testing & Improving

We see how both the Beta Positive and Beta Negative systems perform very well, with nearly or over 50% of winning trades and a very healthy average winning versus losing trades ratio (over 2). Figures 2 and 3 show the System Reports for the Beta Up and Beta Down Systems. We applied Beta Up System to a daily chart of IBM, and the Beta Down System to a daily chart of Arcadia Financial.

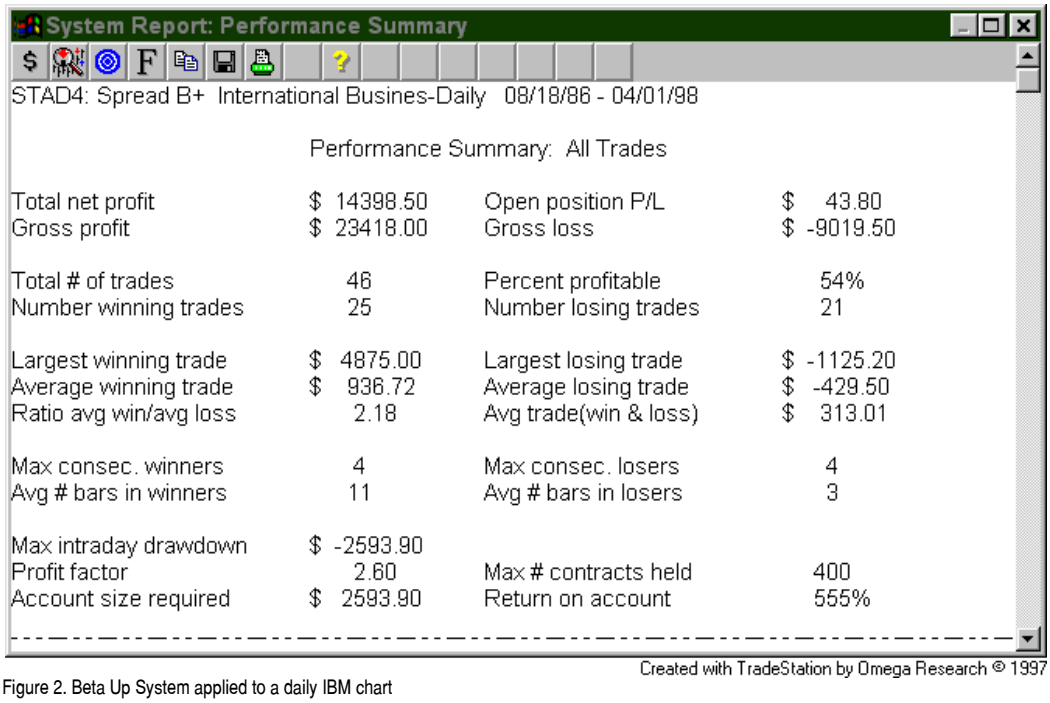


Figure 2. Beta Up System applied to a daily IBM chart

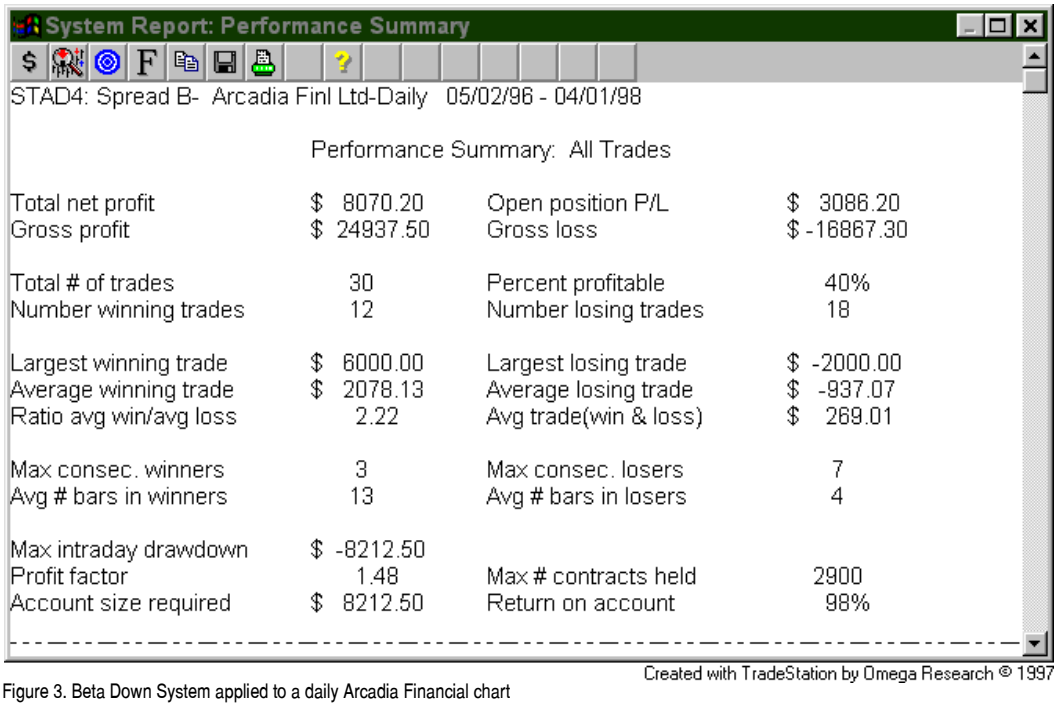


Figure 3. Beta Down System applied to a daily Arcadia Financial chart

These two systems present a unique opportunity. If a basket of beta positive stocks are traded with the B+ system together with a basket of beta negative stocks with the B- system, you will create a portfolio that will tend to perform well whether the market is rising or falling. These two systems, applied to such a group of stocks, should complement each other perfectly. While one of them is standing by waiting for the market to change, the other should be taking profits from the market that is moving in its direction. Adding stocks with both a strong beta positive and a strong beta negative should result in a portfolio with a smooth and constant up-rising equity curve.

Such ideas can be easily tested with Portfolio Maximizer, as it will allow merging of multiple systems applied to multiple securities. Displaying detailed portfolio system results as well as equity and many other analysis. Figure 4 shows the Portfolio Maximizer report for the two systems; the Beta Up system applied to IBM and the Beta Down system applied to AAC:

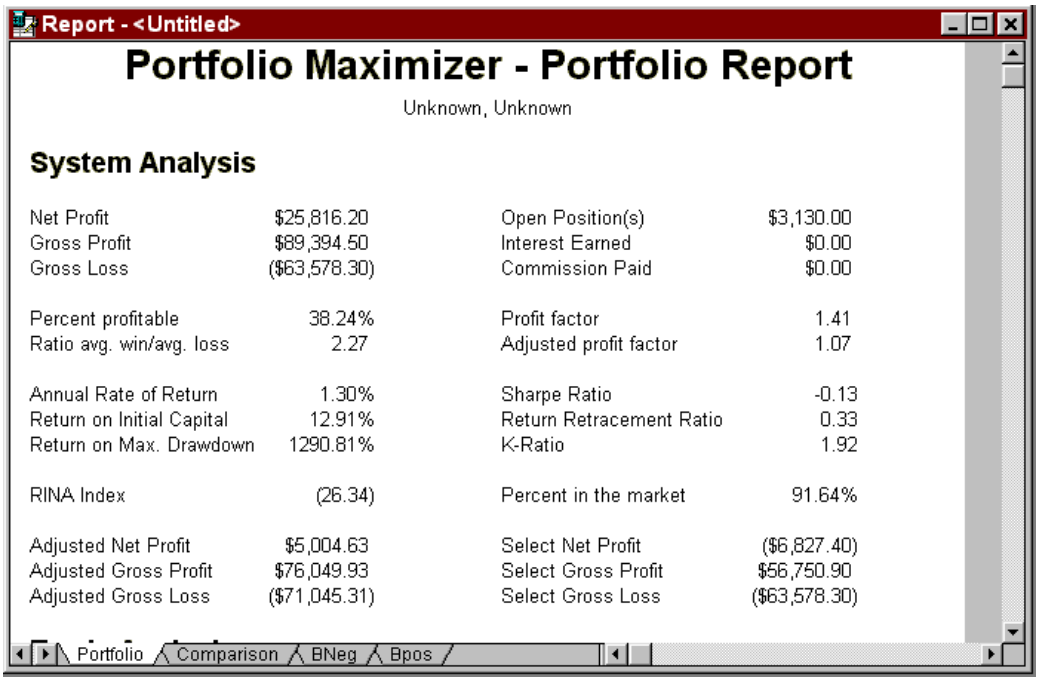


Figure 4. Portfolio Maximizer Report for the two systems

Suggestions for Improvement

You can easily obtain a list of beta-Positive and beta-Negative stocks with TradeStation if you are using the Omega Research Downloader with Dial Data. You will be able to download fundamental information for all you stocks, and two of the available fields are Beta-Up and Beta-Down. Two ShowMe studies named "S_BetaDn > X" and "S_BetaUp > X" (which are shipped with TradeStation and SuperCharts) and the ChartScanner will aid you in this procedure.

This system seems to work very well for most stocks, yet you can use this idea with any timing analysis on both the stock and the index. And again, trading a basket of beta positive and beta negative stocks will allow the equity curve of your portfolio to perform well if the market rises or drops.

First, create a new workspace with one chart with a symbol from the directory or custom security list you want to scan. Then apply both the S_BetaDn > X and the S_BetaUp > X ShowMe Studies. Both studies will have an input called X which you can set to 1. This will produce an alert for any stock with Beta Up or Beta Down greater than one. Save this workspace and access the ChartScanner by using the **Tools - ChartScenner** menu sequence.

Click the **S** button (for ChartScanner setup) and select the workspace you saved with these two ShowMe studies. (Remember to set up the rest of the options of this window. If you need to, refer to your TradeStation User's Manual for information on setting up ChartScanner). Once you've set up your options, click **OK** and then click the **RUN** button one the ChartScanner report window.

Once the ChartScanner is finished, you will have a list of all the securities that are Beta Up and Beta Down. Figure 5 shows the resulting ChartScanner report. You will be able to get a list of

all the Beta Up and Beta Down stocks separately by using the drop down list (on upper right, shown with all the option) on the upper right.

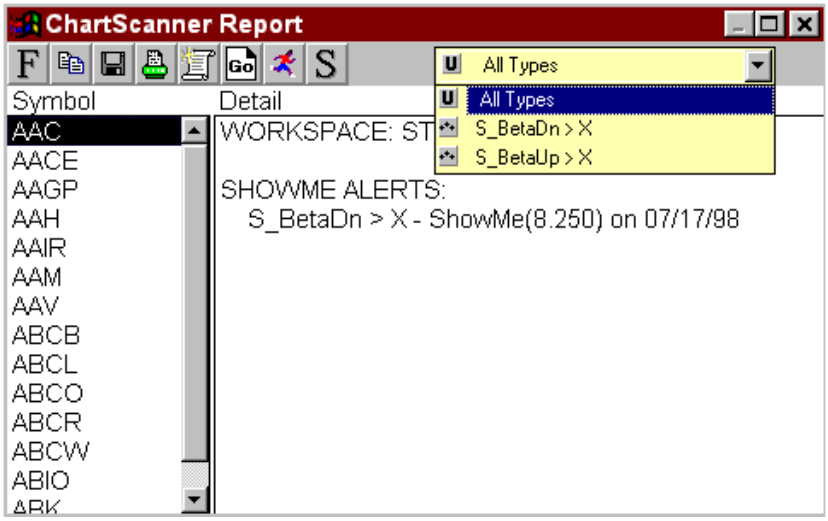


Figure 5 ChartScanner Report listing all securities meeting criteria

CHAPTER 4

Volatility Breakout Systems

Volatile markets are characterized by sharp jumps in price, and volatility breakout systems are designed to take advantage of this type of change in volatility. Volatility breakout systems generally have the following characteristics:

- Substantial amount of time out of the market.
- High percentage of winning trades, but with a small profit per trade.
- Don't take advantage of big moves.
- Exciting to trade because trades are quick and short-term.
- Based solely on price movement.

When designing this type of system, the key is to effectively anticipate and take advantage of a significant change in volatility and then exit the position before a loss of profit. In this chapter, we present two volatility breakout systems, which, like all volatility breakout systems, are designed to capture significant change in volatility and limit losses during directionless and/or trending phases.

In This Chapter

Displaced Moving Average & Volume System

A displaced moving average allows us to shift a moving average forward by a specified number of bars. In other words, rather than plotting the moving average we calculated today on today's price bar, we plot the moving average on a later bar. Displaced means that the moving average is displaced from its conventional bar and placed on a future bar. Displaced moving averages tell us the numeric value a moving average will have on a bar "ahead of time" and help to reduce "whipsaw" losses.

The second component of this system is volume. Volume is the number of shares or contracts that change hands during a specified time period. Daily volume, for example, refers to the number of shares or contracts traded in one day. A market move that is accompanied by an increase in volume is more likely to continue in the same direction than a market move with a decrease in volume. In a healthy uptrend, volume should increase on the up bars and decrease on the down bars; in a robust downtrend, volume should increase on the down bars and decrease on the up bars.

In our Displaced Moving Average and Volume system, we apply a displaced moving average (DMA) to both price and volume. Our buy setup is a close above a DMA of closing prices accompanied by volume that is above a DMA of volume. Our sell setup is a close below a DMA of closing prices accompanied by volume greater than a DMA of volume.

Our entries include a volatility condition. Instead of simply buying at the market on the next open after a setup, we'll require that prices make a move away from the opening price before we buy or sell. To go long, we'll wait for prices to rally to the open plus 20% of the 10-bar average true range. If our entry requirements are not met on the first bar after the setup, we'll pass on the trade. To sell short, we'll wait for prices to decline to the open minus 20% of the 10-bar average true range. If the entry doesn't occur on the first bar after the setup, we'll pass on the trade. Figure 1 shows the system, along with a displaced volume indicator, applied to the chart.

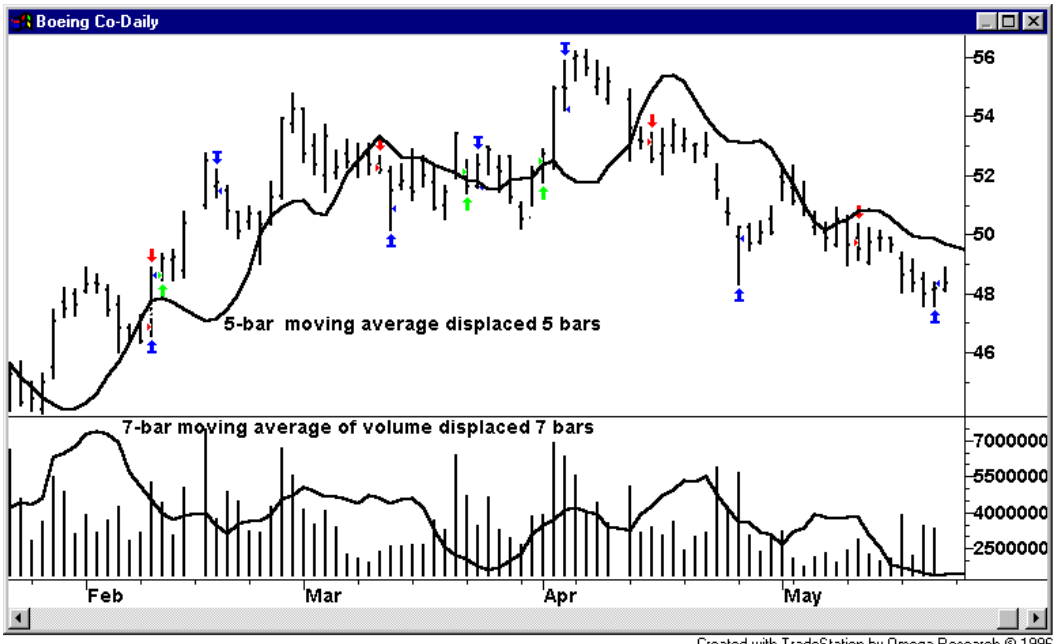


Figure1. Displaced Moving Average and Volume System applied to a daily chart

When we enter a new position, we'll place a money management stop. To lock in profits when prices move in our favor, we'll trail a % risk trailing stop. A % risk trailing stop enables us to specify the percent of the maximum open profit we are willing to give back before our position is closed out. To set a % risk trailing stop, we must also indicate the minimum profit level

(known as a floor) that our position must reach before the % risk trailing stop takes effect. The maximum profit is calculated from our entry point to the highest high if we are long and from our entry point to the lowest low if we are short. Then the percent of this amount that we are willing to risk is subtracted and the trailing stop is set at that point. For example, let's say that we place a 30% risk trailing stop with a $400 floor. When open profits reach the floor value of $400, the stop will become active and will be placed at the maximum profit-to-date minus 30%.

Our exit strategy will be to exit on the close when prices close on the opposite side of the displaced moving average. In other words, we'll exit a long position on a close below the DMA and a short position on a close above the DMA.

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. The long and short entries reverse your position, whereas the exits close out your existing position and exit you from the market. We did some minor setup work, which involved calculating the displaced moving averages. The entry and exits are described next.

Setup

a) Calculate the 7-bar moving average of the closing prices, and displace it by 5 bars.

b) Calculate the 7-bar moving average of the volume, and displace it by 5 bars.

Long Entries

a) When the close crosses above the average of the closing prices, and the volume is greater than the average of the volume, then place a buy stop order at the open plus a certain number of points. The number of points is 20% of the 10-bar average true range.

Short Entries

a) When the close crosses below the average of the closing prices, and the volume is less than the average of the volume, then place a sell stop order at the open minus a certain number of points. The number of points is 20% of the 10-bar average true range.

Exits

a) Place a trailing stop that exits long positions at the lowest low of the last 5 bars.

b) Place a trailing stop that exits short positions at the highest high of the last 5 bars.

c) Exit long positions on the next bar at market when the close crosses below the moving average.

d) Likewise, exit short positions on the next bar at market when the close crosses above the moving average.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Displaced Moving Average & Volume (STAD 4: DMA C/V)

```
Inputs: Price(Close), PLen(7), PDisp(5), VLen(7), VDisp(5), ATRPcnt(.20), Trade(10000), Lot(100);
Vars: DMA(0), DMAV(0);

{Assignment of Displaced Moving Average values to variables}
DMA = Average(Price, PLen)[PDisp];
DMAV = Average(Volume, VLen)[VDisp];
```

```
{Buy Criteria Evaluation}
IF Close Crosses Above DMA AND Volume > DMAV Then
        Buy NumUnits(Trade,Lot) Shares Next Bar at Open Next Bar + (ATRPcnt * AvgTrueRange(10))
        Points Stop;

{Sell Criteria Evaluation}
IF Close Crosses Below DMA AND Volume < DMAV Then
        Sell NumUnits(Trade,Lot) Shares Next Bar at Open Next Bar - (ATRPcnt * AvgTrueRange(10))
        Points Stop;

{Trailing Stops}
ExitLong Next Bar at Lowest(Low, 5) Stop;
ExitShort Next Bar at Highest(High, 5) Stop;

{System Exits}
IF Close Crosses Below DMA Then
        ExitLong Next Bar at Market;
IF Close Crosses Above DMA Then
        ExitShort Next Bar at Market;
```

Inputs

Following is the list of the inputs we used in this system:

Inputs	Default	Description
Price	Close	Price to use when calculating displaced moving average.
PLen	7	Length, expressed in bars, to use to calculate the moving average of the price.
PDisp	5	Length, expressed in bars, by which to displace the moving average of the price.
VLen	7	Length, expressed in bars, to use to calculate the moving average of the volume.
VDisp	5	Length, expressed in bars, by which to displace the moving average of the volume.
ATRPcnt	.20	Percentage of the average true range to add to the open as points to determine the buy and sell entry price.
Trade	10,000	Dollar amount to invest per transaction.
Lot	100	Minimum shares to trade per transaction.

In addition to these inputs, we identified the following variables:

```
Vars: DMA(0), DMAV(0);
```

Setup

First we calculate the displaced moving averages for both the close and the volume. We store the displaced moving average of the closing prices in the variable DMA. We store the displaced moving average of the volume in the variable DMAV.

```
DMA = Average(Price, PLen)[PDisp];
DMAV = Average(Volume, VLen)[VDisp];
```

Long Entries

When the close price crosses above DMA and the volume is greater than DMAV, then we'll place a buy stop order. The number of shares to buy is determined using the NumUnits function. Please refer to the section in Chapter 2, titled, "Investing a Fixed Dollar Amount," for information on using the NumUnits function. The entry price is determined by adding 20% of the 10-bar average of the true range to the open price.

```
IF Close Crosses Above DMA AND Volume > DMAV Then
        Buy NumUnits(Trade,Lot) Shares Next Bar at Open Next Bar +
        (ATRPcnt * AvgTrueRange(10)) Points Stop;
```

Short Entries

When the close price crosses below DMA and the volume is less than DMAV, then we'll place a sell stop order. Again, the number of shares to sell is determined using the NumUnits function. Please refer to the section in Chapter 2, titled, "Investing a Fixed Dollar Amount," for information on using the NumUnits function. The entry price is determined by subtracting 20% of the 10-bar average of the true range from the open price.

```
IF Close Crosses Below DMA AND Volume < DMAV Then
        Sell NumUnits(Trade,Lot) Shares Next Bar at Open Next Bar -
        (ATRPcnt * AvgTrueRange(10)) Points Stop;
```

Exit Orders

To exit, we'll look for the 5-bar extreme. We'll exit long positions when the price reaches the lowest low of the last 5 bars. Likewise, we'll exit short positions when the price reaches the highest high of the last 5 bars.

```
ExitLong Next Bar at Lowest(Low, 5) Stop;
ExitShort Next Bar at Highest(High, 5) Stop;
```

We'll also exit long positions when the close crosses below DMA; we'll exit on the next bar at market. We'll also exit short positions on the next bar at market when the close crosses above DMA.

```
IF Close Crosses Below DMA ThenExitLong Next Bar at Market;
IF Close Crosses Above DMA ThenExitShort Next Bar at Market;
```

General System Format

When we apply a system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified $10,000 as the trade size and 100 shares as the lot size.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

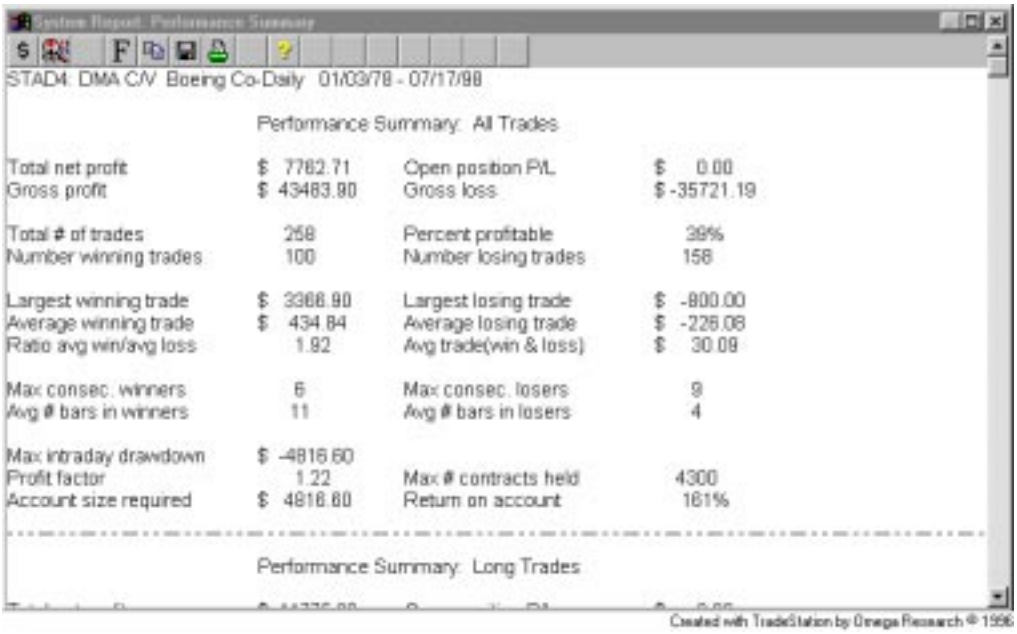
Under the **Stops** tab, we enabled a % risk trailing stop. This type of stop enables you to indicate what percent of the maximum open profit you are willing to give back before the position is automatically closed out. The % risk trailing stop also requires that you specify a minimum profit level that must be reached before the stop takes effect.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the **Do not allow multiple entries in the same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 2 shows a sample System Report for the Displaced Moving Average and Volume System.



STAD4: DMA C/V Boeing Co-Daily 01/03/78 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 7762.71	Open position P/L	$ 0.00
Gross profit	$ 43483.90	Gross loss	$ -35721.19
Total # of trades	258	Percent profitable	39%
Number winning trades	100	Number losing trades	158
Largest winning trade	$ 3366.90	Largest losing trade	$ -800.00
Average winning trade	$ 434.84	Average losing trade	$ -226.08
Ratio avg win/avg loss	1.92	Avg trade(win & loss)	$ 30.09
Max consec. winners	6	Max consec. losers	9
Avg # bars in winners	11	Avg # bars in losers	4
Max intraday drawdown	$ -4816.60		
Profit factor	1.22	Max # contracts held	4300
Account size required	$ 4816.60	Return on account	161%

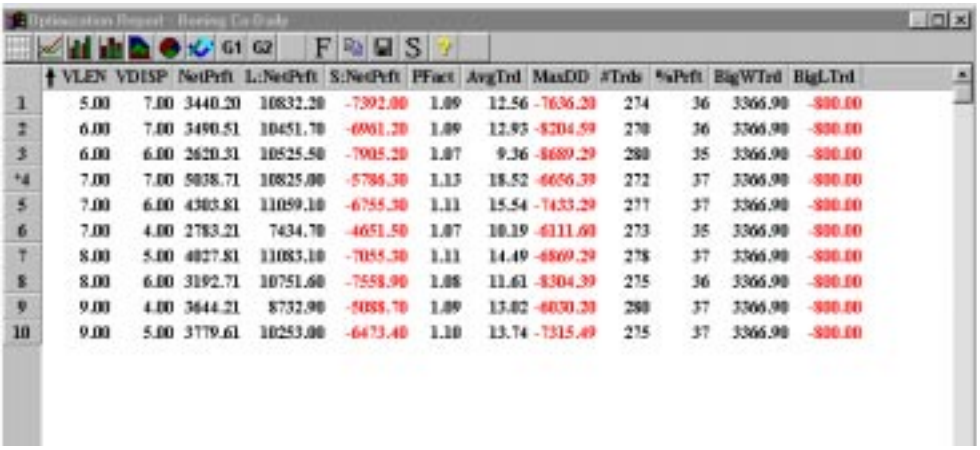
Performance Summary: Long Trades

Created with TradeStation by Omega Research © 1998

Figure 2. Sample System Report for the Displaced Moving Average and Volume System

Trading both the long and short sides of the market, we are able to produce only 39% of profitable trades. Still, since most stock traders (for the last 20 years especially) have focused their efforts on going long and avoiding the short side of the market, our results are promising. The ratio of the average win to the average loss was 1.92 and the profit factor (the number of dollars won for every dollar lost) was 1.22. The system let profits run by staying in the winning trades for 11 bars and cut losses short by staying in the losing trades for only four bars.

We optimized for the length of the moving average of volume and for the number of bars forward the average is displaced. Figure 3 shows the resulting Optimization Report.



	VLEN	VDISP	NetPrft	L:NetPrft	S:NetPrft	PFact	AvgTrd	MaxDD	#Trds	%Prft	BigWTrd	BigLTrd
1	5.00	7.00	3440.20	10832.20	-7392.00	1.09	12.56	-7636.20	274	36	3366.90	-800.00
2	6.00	7.00	3490.51	10451.70	-6961.20	1.09	12.93	-8204.59	270	36	3366.90	-800.00
3	6.00	6.00	2620.31	10525.50	-7905.20	1.07	9.36	-8689.29	280	35	3366.90	-800.00
*4	7.00	7.00	5038.71	10825.00	-5786.30	1.13	18.52	-6656.39	272	37	3366.90	-800.00
5	7.00	6.00	4303.81	11059.10	-6755.30	1.11	15.54	-7433.29	277	37	3366.90	-800.00
6	7.00	4.00	2783.21	7434.70	-4651.50	1.07	10.19	-6111.60	273	35	3366.90	-800.00
7	8.00	5.00	4027.81	11083.10	-7055.30	1.11	14.49	-6869.29	278	37	3366.90	-800.00
8	8.00	6.00	3192.71	10751.60	-7558.90	1.08	11.61	-8304.39	275	36	3366.90	-800.00
9	9.00	4.00	3644.21	8732.90	-5088.70	1.09	13.02	-6030.20	280	37	3366.90	-800.00
10	9.00	5.00	3779.61	10253.00	-6473.40	1.10	13.74	-7315.49	275	37	3366.90	-800.00

Figure 3. Optimization Report for length of displacement

The best results were obtained with a 7-bar average displaced forward 7 bars. This combination produced the best net profit, profit factor, and average trade, and the lowest maximum drawdown. The bar graph shown in Figure 4 illustrates the effect on the size of the average trade varying the length of the moving average of volume from five to nine.

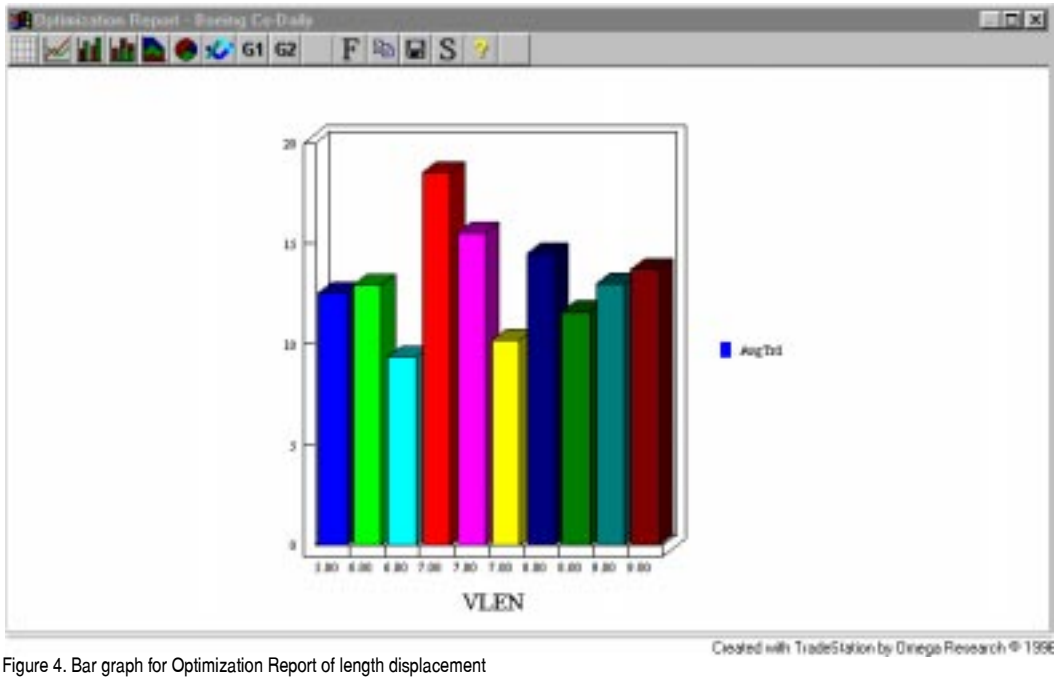


Figure 4. Bar graph for Optimization Report of length displacement

Suggestions for Improvement

We can almost certainly improve the performance of this system by optimizing the length and the displacement of the moving average of closes. Another idea would be to average a price other than the closing price. For example, would basing the moving average on the midpoints of the bars be better than averaging the closes? Would it make sense for this system to incorporate two displaced moving averages of price — one of the highs and the other of the lows? In system development there is almost always something else to consider and to test, some new possibility to be explored.

Price Distribution System

Two concepts commonly used in statistics when referring to distribution of data are *skewness* and *Kurtosis*. The concepts can be defined as follows:

Skewness: Returns the skewness of a distribution. Skewness characterizes the degree of asymmetry of a distribution around its mean. Positive skewness indicates a distribution with an asymmetric tail extending toward more positive values. Negative skewness indicates a

distribution with an asymmetric tail extending toward more negative values. Figure 5 illustrates this concepts:

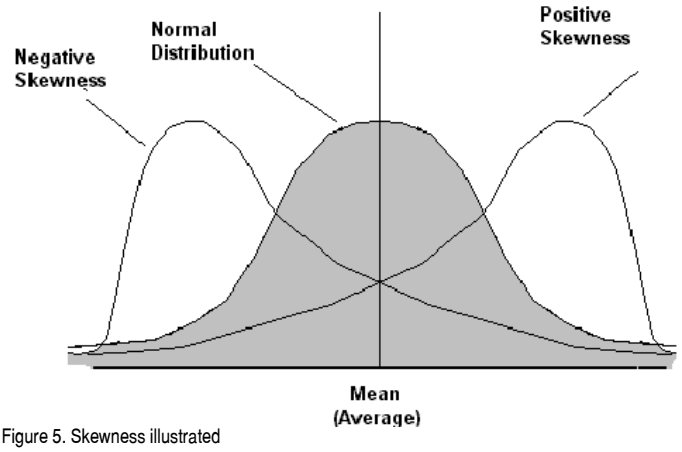


Figure 5. Skewness illustrated

Kurtosis: Kurtosis characterizes the relative peakedness or flatness of a distribution compared with the normal distribution. Positive kurtosis indicates a relatively peaked distribution. Negative kurtosis indicates a relatively flat distribution. Figure 6 illustrates this concept:

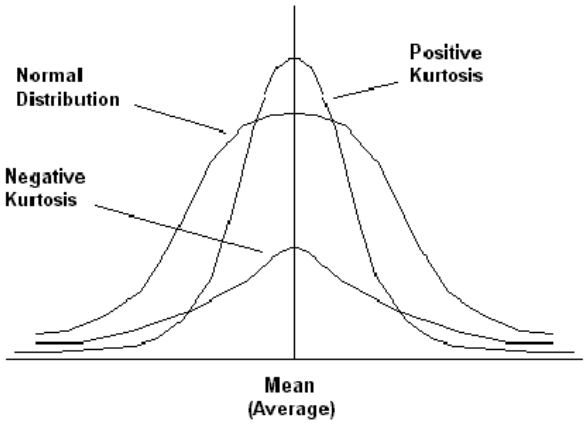


Figure 6. Kurtosis illustrated

We can use Kurtosis to determine if the market is trending or choppy. When we have a normal or peaked distribution (Kurtosis greater than 0) then we can assume that the market is going sideways. When the distribution of prices is starting to flatten, that is a sign that the market is starting to move directionally, so we can look to establish a position. When applying Kurtosis to a distribution of prices, it will work in a similar fashion to the ADX; it will tell you whether or not the market is moving sideways, but it will not tell you the direction.

Therefore, we will use skewness to determine in which direction the market is *leaning*. If the distribution curve is displaced to the right (positive skewness), it would mean that the higher prices have dominated. If the distribution curve is displaced to the left (negative skewness) then we can assume that lower prices are dominating.

If we draw the distribution of prices of both a sideways and a directional market, we will see that generally the sideways market will tend to have distribution that approaches a normal distribution, or even a distribution with a positive Kurtosis; the concept of sideways market is that the prices are condensing into a defined area and have market support and resistance zones (the support and resistance zones would be the tails of the distribution). If a market is moving

significantly in one direction, we can assume that the distribution will be much flatter than normal (or negative Kurtosis) and that the prices will not linger in the same area for much time. Figure 7 illustrates this concept:

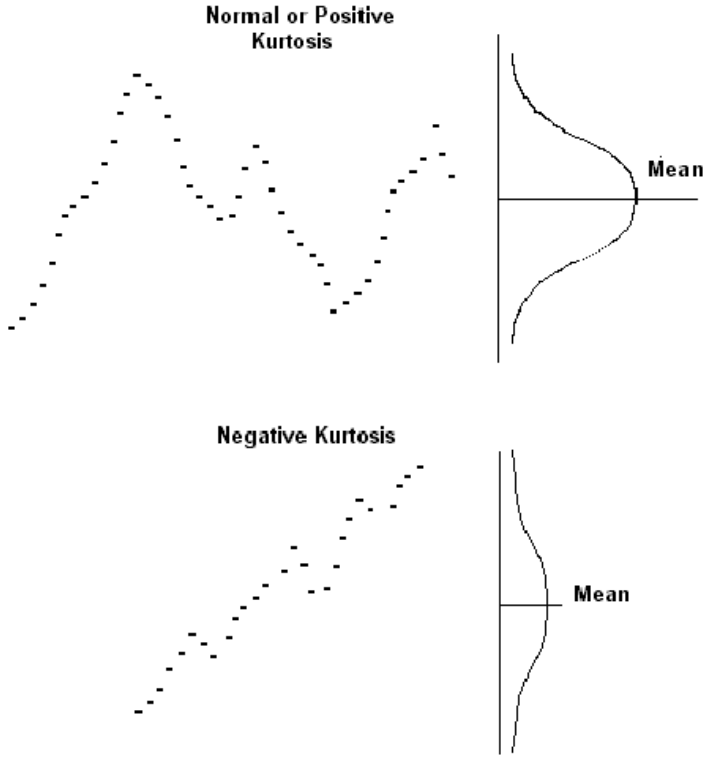


Figure 7. Positive and negative Kurtosis

So one way of trying to find the change in mode of the market is to follow the Kurtosis of the last 40 closes. If the Kurtosis is positive, it probably means that the market is stagnant, and is moving sideways. When Kurtosis crosses under zero, we can assume that the market is starting to move in some direction, so we can use the skew to determine the direction in which the market is thinner (has more support or resistance).

Our system will look for Kurtosis to cross under zero, and then if the skew is positive, we will establish a long position. If the skew is negative, we will establish a short position.

To exit, we will apply a trailing stop to for both sides: once we enter the market on the buy stop, we'll calculate the 4-bar average of the range and subtract this value from the low. This will be our long exit price for bar 2. On the subsequent bar, we'll get the exit price calculated in the last point and add to it a third of the difference between the low and the previous stop price. Repeating this operation at the end of every bar will give us our exit price for bar 3 and beyond.

Likewise, once we enter the market on the sell stop, we'll calculate the 4-bar average of the range and add this value to the high. This will be our short exit price for bar 2. On the subsequent bar, we'll get the exit price calculated in the last point and subtract from it a third of the difference between the previous stop price and the high. Repeating this operation at the end of every bar will give us the exit price for bar 3 and beyond.

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. The long and short entries reverse your position, whereas the exits close out your existing position and exit you from the market. We also performed some setup work, which involved identifying the Inside Day and subsequent breakouts. The setup, entry and exits are described next.

Setup

a) Calculate Kurtosis and Skew.

Long Entries

a) Check for Kurtosis to cross under zero. If the skew is positive, we will establish a long position.

Short Entries

a) Check for Kurtosis to cross under zero. If the skew is negative, we will establish a short position.

Exit Orders

a) Once the bar of entry of a long position has closed, calculate the 4-bar average of the range and subtract this value from the low. This is our long exit price for bar 2.

b) On the subsequent bar, get the exit price calculated in the last point and add to it a third of the difference between the low and the previous stop price. Repeating this operation at the end of every bar gives us our exit price for bar 3 and beyond.

c) Once the bar of entry of a short position is closed, calculate the 4-bar average of the range and add this value to the high. This is our short exit price for bar 2.

d) On the subsequent bar, we'll get the exit price calculated in the last point and subtract from it a third of the difference between the previous stop price and the high. Repeating this operation at the end of every bar gives us the exit price for bar 3 and beyond.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Price Distribution (STAD 4: PriceDist)

```
Inputs: KurtLen(40), SkewLen(15);
Vars:  MP(0), StopPrice(0), Kurt(0), Skewv(0);

Kurt = Kurtosis(Close,KurtLen);
Skewv = Skew(Close,SkewLen);

{Entry orders}
If Kurt crosses under 0 then Begin
        If Skewv > 0 then
                Buy this bar at close;
        If Skewv < 0 then
                Sell this bar at close;
End;

{ Trailing stops }
MP = MarketPosition;
If MP = 1 and MP[1] <> 1 thenStopPrice = Low - Average(Range,4);

If MP = -1 and MP[1] <> -1 thenStopPrice = High + Average(Range,4);

If MP = 1 then Begin
        Exitlong next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

```
If MP = -1 then Begin
        Exitshort next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

Inputs

Following is the list of the inputs we used in this system:

Inputs	Default	Description
KurtLenNumber	40	Number of bars used to calculate Kurtosis
SkewLenNumber	15	Number of bars used to calculate skew

In addition to these inputs, we identified the following variables:

```
Vars:  MP(0), StopPrice(0), Kurt(0), Skewv(0);
```

Setup

The formulas for Kurtosis and skew are provided on your CD as functions, Kurtosis and Skew, respectively. We begin by using the functions to calculate Kurtosis and skew and store the values in variables:

```
Kurt = Kurtosis(Close,KurtLen);
Skewv = Skew(Close,SkewLen);
```

Long & Short Entries

If Kurtosis crosses under zero, then we will assume that the market is changing from choppy to directional, so we will look to establish a position; we will establish a long position if the Skew of the prices is positive and a short position if the skew of the distribution of prices is negative.

```
If Kurt crosses under 0 then Begin
        If Skewv > 0 then Buy this bar on Close;
        If Skewv < 0 thenSell this bar on Close;
End;
```

Exit Orders

We will use a version of the trailing stops that we have used in previous volumes of the STAD Club. We will use the variable MP to store the value of the keyword MarketPosition. We do this so that we can reference the values of MarketPosition of previous bars.

```
MP = MarketPosition;
```

If we are in a long position, and one bar ago we were not in a long position, then we will assign the low of the current bar minus the 4-bar average range to the variable StopPrice.

```
If MP = 1 and MP[1] <> 1 then
        StopPrice = Low - Average(Range,4);
```

If we are in a short position, and one bar ago we were not is a short position, then we will assign the high of the current bar plus the four bar average of the range to the variable StopPrice

```
If MP = -1 and MP[1] <> -1 then
        StopPrice = High + Average(Range,4);
```

If we are in a long position, we will place a stop order to exit from the position at the StopPrice or anything lower. Then we will calculate a third of the distance between the low of the current bar and the StopPrice and we will add that from the current stop price, the resulting value will be the new StopPrice.

```
If MP = 1 then Begin
        Exitlong next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

If we are in a short position, we will place a stop order to exit from the position at the StopPrice or anything higher. Then we will calculate a third of the distance between the the StopPrice and the high of the current bar and we will subtracat that from the current stop price, the resulting value will be the new StopPrice.

```
If MP = -1 then Begin
        Exitshort next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we normally use the options in the **Format** dialog box to format costs, stops, and properties. However, in this system we did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded. We specified 1 as the default contract size to trade.

Note: *Remember that Commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling.*

Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or a dollar amount per contract/share you want to risk before exiting from the position.

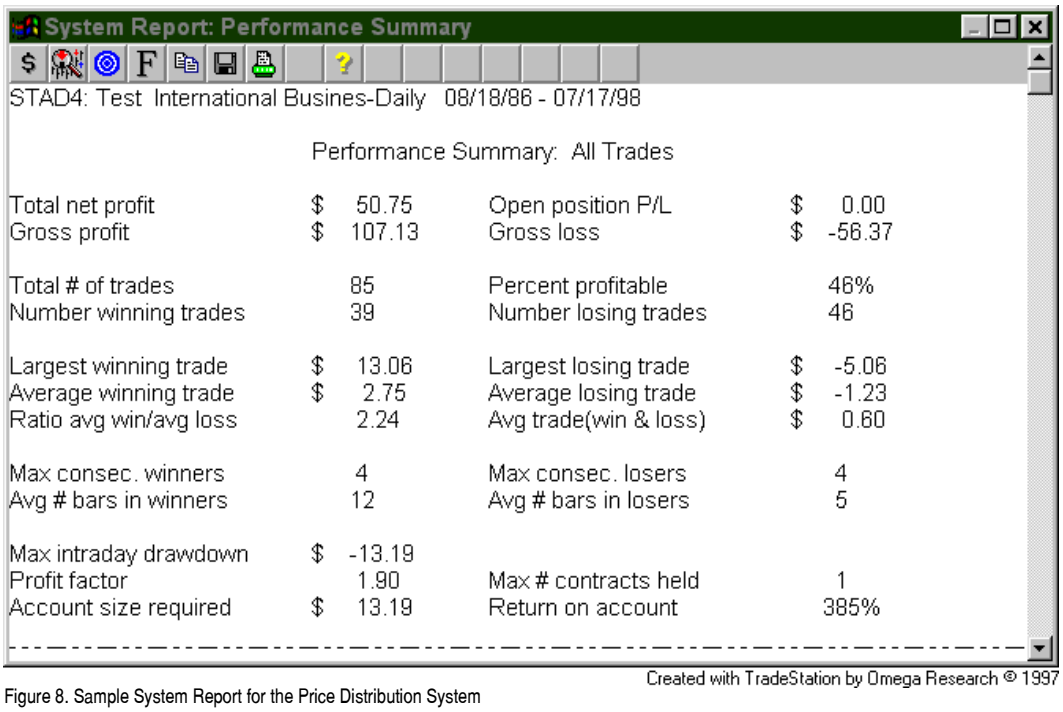
Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited.When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

In the **Properties** tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a long position, and market conditions generate another long entry order, the order is ignored.

Testing & Improving

As you can see from Figure 8, this system has the typical results of a support and resistance system. This system identifies areas of strong support and resistance levels (choppy segments) by following the values of kurtosis, and when these levels start to weaken we will attempt to establish a position in the direction which the market is moving. Knowing all this, we would

probably want to improve on the percentage of winning trade, probably by improving how the system picks the direction in which to enter the market.



System Report: Performance Summary

STAD4: Test International Busines-Daily 08/18/86 - 07/17/98

Performance Summary: All Trades

Total net profit	$ 50.75	Open position P/L	$ 0.00
Gross profit	$ 107.13	Gross loss	$ -56.37
Total # of trades	85	Percent profitable	46%
Number winning trades	39	Number losing trades	46
Largest winning trade	$ 13.06	Largest losing trade	$ -5.06
Average winning trade	$ 2.75	Average losing trade	$ -1.23
Ratio avg win/avg loss	2.24	Avg trade(win & loss)	$ 0.60
Max consec. winners	4	Max consec. losers	4
Avg # bars in winners	12	Avg # bars in losers	5
Max intraday drawdown	$ -13.19		
Profit factor	1.90	Max # contracts held	1
Account size required	$ 13.19	Return on account	385%

Created with TradeStation by Omega Research © 1997

Figure 8. Sample System Report for the Price Distribution System

Aside from this, the system appears to have a very strong ratio of average winning versus losing trades, and essentially overall good results.

Suggestions for Improvement

In order to further filter out bad signals, you can change the entry orders to stop orders. This way you can establish a long or short position only when the market follows through. This should improve the percentage of wining trades, and somewhat (maybe) reduce the drawdown.

Alternatively, instead of looking for skew of the distribution to determine the direction of the market, you can place a long entry stop order above the market (at the highest high of the last 6 bars) and a short entry stop order under the market. This way you will let the market tell you in what direction it is moving.

Additional Formulas

Kurtosis

The formula for the Kurtosis function is quite complex. The mathematical formula is as follows:

$$\left\{\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum\left(\frac{x_j-\bar{x}}{s}\right)^4\right\}-\frac{3(n-1)^2}{(n-2)(n-3)}$$

The Kurtosis function is included on your CD and its EasyLanguage is described here. We will use two inputs:

Price	Value on which Kurtosis will be calculated
Length	Number of points (bars) to be included in the calculation of Kurtosis

The EasyLanguage is as follows:

```
Input: Price(numericSeries), Length(numeric);
Vars: P1(0), P2(0), P3(0), Avg(0), Std(0);
```

The following EasyLanguage instruction represents the first section of the formula:

$$\frac{n(n+1)}{(n-1)(n-2)(n-3)}$$

```
P1 = Length*(Length+1)/((Length-1)*(Length-2)*(Length-3));
```

The following EasyLanguage instruction represents the third section of the formula:

$$-\frac{3(n-1)^2}{(n-2)(n-3)}$$

```
P3 = 3 * Square(Length-1)/((Length-2)*(Length-3));
```

We calculate the average and the standard deviation of the last Length elements:

```
Avg = Average(Price, Length);
Std = StdDevS(Price, Length);
```

Finally, we will calculate the second section of the formula. This involves a summation of the 4^{th} power of each of the prices minus the average divided by the standard deviation. Here is the formula and its EasyLanguage:

$$\sum\left(\frac{x_j - \bar{x}}{s}\right)^4$$

```
P2 = 0;
For value1 = 0 to Length-1 Begin
        P2 = P2 + Power((Price[value1]-Avg)/Std, 4);
End;
Kurtosis = P1 * P2 - P3;
```

Skew

The formula for skew is another complex statistical expression, following are both the math and EasyLanguage versions:

$$\frac{n}{(n-1)(n-2)}\sum\left(\frac{x_j - \bar{x}}{s}\right)^3$$

We will use two inputs:

Price	Value on which Kurtosis will be calculated
Length	Number of points (bars) to be included in the calculation of Kurtosis

```
Inputs: Price(Numeric), Length(Numeric);
```

Additionally, we will use the following variables in this function

```
Vars: Summ(0), Avg(0), Std(0), Y(0);
```

We calculate the Average and the standard deviation of the last Length elements.

```
Avg = Average(Price, Length);
Std = StdDevS(Price, Length);
```

The following EasyLanguage line calculates this first section of the Skew formula:

$$\frac{n}{(n-1)(n-2)}$$

```
Y = Length / ((Length-1) * (Length-2));
```

The following for-loop will calculate this part of the statistical function:

$$\sum\left(\frac{x_i - \bar{x}}{s}\right)^3$$

```
Summ = 0;
For value1 = 0 TO Length-1 Begin
        Summ = Summ + Power((Price[value1] - Avg) / Std ,3);
End;
Skew = Y * Summ
```

Standard Deviation

Note that we use the StdDevS function (provided with version 4 of STAD Club) instead of the StdDev function provided with TradeStation. StdDev() calculates the standard deviation of a population, also referred as the “n” or the “biased” method. Kurtosis traditionally uses the “unbiased” or “n-1” method for its calculations, this method is known as the standard deviation using a sample, and we provide it through the StdDevS function. Following is the formula for both functions:

Standard Deviation of a sample

$$\sqrt{\frac{\sum x^2 - \left(\sum x\right)^2}{(n-1)}}$$

Standard Deviation of a population

$$\sqrt{\frac{n\sum x^2 - \left(\sum x\right)^2}{n^2}}$$

Following is the EaysyLanguage function for Standard Deviation of a sample. Note that it is identical to the Standard Deviation of a population provided with TradeStation (StdDev), except that it divides the result into (Length - 1) instead of just dividing it into (Length):

```
Inputs : Price(NumericSeries),Length(NumericSimple);
Vars   : SumSqr(0),Avg(0),Counter(0);

If Length <> 0 then Begin
        Avg = Average(Price,Length);
```

```
                SumSqr = 0;
                For counter = 0 to Length - 1 Begin
                        SumSqr = SumSqr + ( Price[counter] - Avg ) * ( Price[counter] - Avg );
                        End;
                StdDevS = SquareRoot( SumSqr / ( Length - 1 ) );
End;
Else
                StdDevS = 0;
```

INDEX

T

V

W