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 the Club

Now that you've determined that system trading is the way to go—that most successful traders are system traders—where do you start? How do you go from having some ideas to actually creating a system?

That's where the *Omega Research System Trading and Development Club* comes in.

This first volume, Volume I, of the *Omega Research System Trading and Development Club* contains 10 systems we've created to help you get started developing your own systems. These systems incorporate popular concepts such as moving average crossovers, pivot points and spreads. We've identified the most common problems with these types of systems and identified techniques to overcome them. By reviewing and testing these 10 systems, you'll be able to understand these techniques and use them or a variation in your own systems.

To help you do that, 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.

When you join the *Omega Research System Trading and Development Club*, you'll receive an additional volume approximately every 2 months, with 10 new systems and this comprehensive guide to help you understand and get the most out of the systems.

***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:

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

Is this a Beginner's Book?

You'll get the most out of this book if you are already familiar with EasyLanguage to some degree. This book is intended for the TradeStation user who has already started writing systems—perhaps the systems aren't as successful as you'd like them to be, or maybe they are but you'd really like some new ideas or techniques to try.

But if you are a beginner, don't worry! You can still apply the systems, study them either on paper or in the PowerEditor, and begin to understand what system development is all about. And most importantly, you'll begin to understand where system ideas come from and to get started developing them. You may also want to consider taking advantage of some of the additional EasyLanguage and system development educational services we offer.

If you are on the other side of the spectrum and are an experienced system developer, what you'll get out of this book is ideas. Perhaps we accomplished a task using a slightly different technique than you did, or in a way that you hadn't thought of. You can use the techniques in this club and apply them as is or somewhat differently, or you can vary and mold them into something even more useful for you.

Additional Educational Services

Omega Research is committed to enhancing individual trading potential through quality education. To help you 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 U.S. 305-551-9991) and ask about the following educational services.

Video Course

We offer a comprehensive video course that introduces you to EasyLanguage, starting with the very basics and taking you through writing actual trading systems. It includes a follow-along workbook that contains exercises and real-life examples and applications.

Once you finish this course, you'll be able to write your own indicators, ShowMe and PaintBar studies, and trading systems. This course is perfect for those who want to learn EasyLanguage at their own pace.

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.

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 *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**.

Join the Club

If you haven't joined the *Omega Research System Trading and Development Club*, don't delay! When you join the club, every two months you'll receive ten new systems along with the comprehensive handbook explaining in detail each system and the process used to develop it. Keep in mind that the club is risk free—if you're not completely satisfied with any of the volumes you receive, simply return it for an immediate refund. You can cancel your membership at any time without further obligation.

To become a charter member of the club and radically increase your ability to design and develop trading systems, call Omega Research at 800-439-7995 (outside the U.S. 305-551-9991).

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 **JANFEB98.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**.

You can now open the systems and indicators 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 the *TradeStation User's Manual*.

CHAPTER 1

General System Development Concepts

When you are developing your systems, there are certain rules of thumb that you should follow. These are guidelines that our users have discovered through years of trial and effort, and we are providing them for you here. For example, you should know that there are three basic types of markets—*trending*, *volatile* and *directionless*—and no one system can work well in all three. The key is to develop a system that works well in one market type and limits your losses in the other market types. This is a basic but very important concept.

Another concept is that you don't need to look at the system as a whole at the beginning. For example, you can have a great idea for an entry signal and really no idea as to how you will want to exit. That doesn't mean you can't start writing a great system. It means you can start by developing your entries until they work really well, then, once that's done, start working on the exits. Maybe you even have several favorite exits that you try with your different entries.

The point is, before you begin developing your systems, read this chapter—it could save you hours of work and headaches!

In This Chapter

Select your Market and System Type

The very first step in developing a system is to decide on the market type you want to trade. This is an important decision because it determines the type of system you will be developing. This section will help you to understand some of the conditions that can occur in the different types of markets and the types of systems that complement those markets. Once you are familiar with the basic system types, you will be able to select the one that you want to use.

Generally, there are three types of markets. The three market types, or phases, are derived from three distinct chart patterns that appear when there is a shift in market action. The phases are *trending*, *volatile*, and *directionless*, and each can be characterized by specific price activity. Each of the three types of markets are tradable, but with markedly different trading systems. Let's take a look at each type of market behavior and the systems that are appropriate to that type of market.

Trending Market

Trending markets are characterized by a large sustained increase or decrease in price. Figure 1-1 shows an example of a trending market.

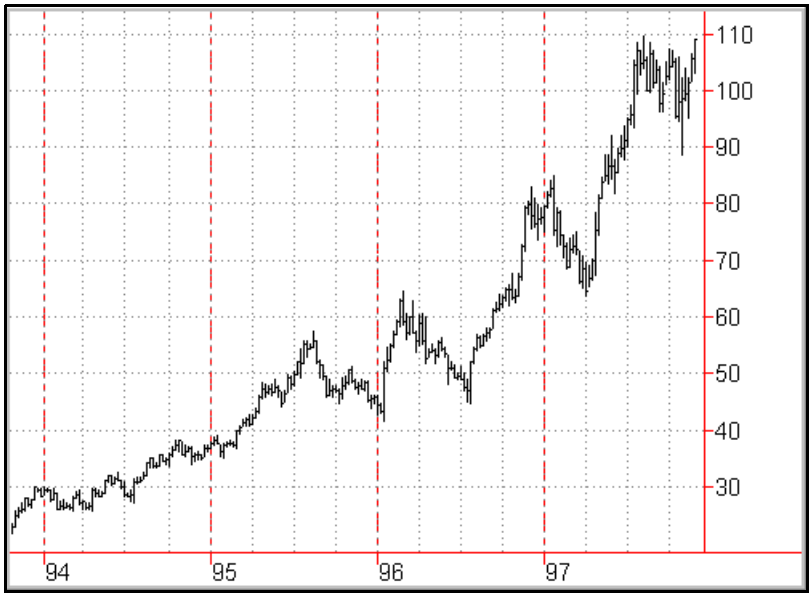


Figure 1-1. Sample of trending market

This market has been in an up-trend since before 1994. The price has almost tripled over this three-year period. This trending market was characterized by sustained up moves with small, short-lived corrections.

Trending Systems

Like the name implies, trending systems are designed for trending markets and to take advantage of all the big trending moves that may occur. In creating a trending system, the number one priority should be that the system never miss a big move. The easiest way to accomplish this is to always be in the market, that is, to always be either short or long. If you always have a position, you will always be there when the big move takes place.

The other method is to always have a "stop" order in the market, resting either above or below the current price (this is the same order as a stop loss, but it is used to enter the market rather than exit). Using a stop to enter the market will protect you because if the market moves quickly in either direction you will be stopped in before the big move begins.

Keep in mind that trending systems tend to lose money in choppy, or directionless phases of the market. They have a small percentage of winning trades, that is, they make their money in a few big trades. This means that if you miss a big move, you may not have enough capital to hold out through the drawdown as you wait for the next big move.

Another design characteristic should be to limit your losses during the market's sideways mode. Remember, no system will make money in every market condition. If the system is designed to make money in a trending market, it will lose money in the choppy phase. Your priority should be to minimize the losses in the directionless market.

Many trending systems make their money in one or two months of the year and break even or lose money for the rest. The most common indicator used in trend following systems is moving averages, most often two—a short moving average and a longer moving average.

Trending Systems have the following characteristics:

- ✓ They make 80% of their profits on 20% of their trades. This is the difficult part of trend trading—a low percentage of winning trades. You need a lot of positive self-esteem and confidence in your abilities to trade a system that loses money on 60 or 65% of its trades. You should also be able to sit through significant drawdown as the market drifts through a directionless period.
- ✓ Many researchers have estimated that any market is in the trend mode 15% of the time and is directionless 85% of the time. A trend following system then, by definition, has a low percentage of profitable trades. A trend following system is psychologically difficult to trade, but if you think you can successfully trade without constant positive feedback, it can prove to be very profitable.

Trending systems are the most popular type of system. With a high percentage of losing trades, you might be wondering why is it so popular. Very simply, trend-following systems can be very profitable overall, and this is why people attempt to trade them. Another reason is that people like to follow (and make money on) the big trends. It is human nature to want to cash in on the big moves in the market. It is innately satisfying to get in early on a trend and watch your profits soar.

Directionless Market

A directionless market is characterized by smaller, insignificant up and down movements in price, with the general movement sideways. The key is that the up and down movements are insignificant. Figure 1-2 shows a chart exhibiting sideways movement.

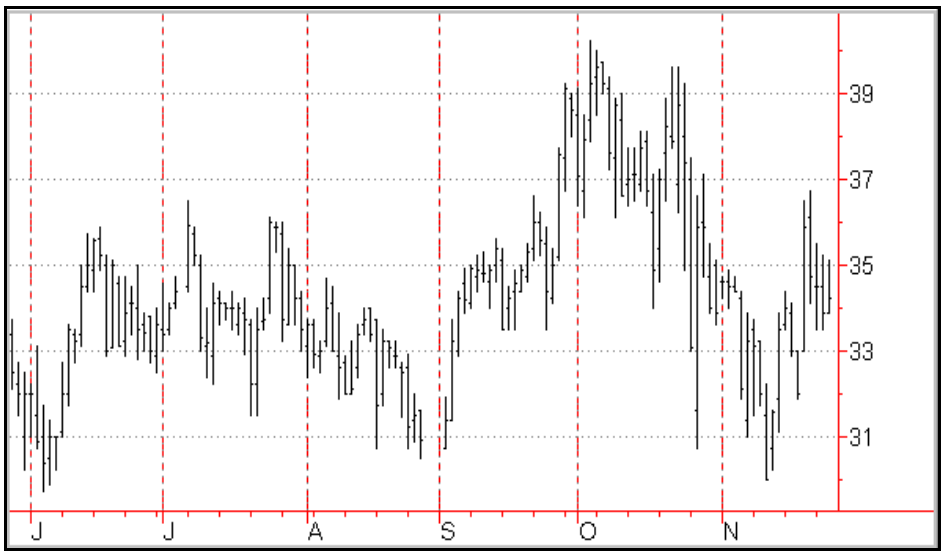


Figure 1-2. Sample directionless market

Support & Resistance Systems

The main focus of a Support & Resistance system is to profit from the price swings that occur in directionless markets. Support and resistance systems start with the premise that markets are directionless 85% of the time. The system attempts to take advantage of this price movement and catch the small swings that take place in sideways or choppy markets. This type of system has a higher number of winning trades, with small profits on each trade. It misses the full trend because it exits early in the trend move as the market becomes quickly overbought or oversold.

An Support & Resistance system is built on the premise of buying low and selling high. As you are buying when prices are low and selling when prices go up, you are actually trading against the trend. Essentially, you are picking tops and bottoms. You buy low and sell high, but the market keeps going higher. You keep selling as the market goes higher, and keep taking small losses until the market finally turns down and gives you a profitable trade.

Although an Support & Resistance system is easier to trade emotionally, many traders don't trade this type of system because they miss the big move. The drawback of support and resistance systems is that they usually have small profits and larger losses, losing money as the market trends; the system keeps shorting a market that is in an up-trend or buying a market that is a down-trend.

Volatile Market

A volatile market is characterized by sharp jumps in price. Figure 1-3 contains a volatile market. Notice the opening gaps.

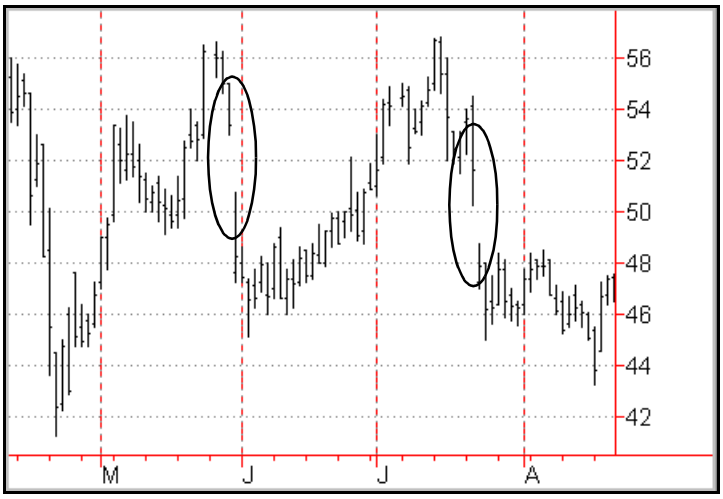


Figure 1-3. Sample volatile market.

Volatility Expansion System

Volatility Breakout systems are designed to take advantage of the sharp jumps in price characteristic of volatile markets. Volatility breakout systems profit from market action like the movement depicted in the chart in Figure 1-3. Basically the system measures recent volatility, and attempts to trade an immediate increase by buying an upside breakout with increased volatility or selling a downside breakout as the volatility increases. Another measure of increasing volatility might be the difference or spread between two moving averages, gap openings, or an increase in the daily range.

The trades generated by volatility breakout systems are usually short term, and when trading this system you should be out of the market a significant amount of time. Volatility expansion systems generate a high percentage of winning trades, with small profits per trade. Most traders characterize the S&P futures market as "volatile" and neither trending systems nor support & resistance systems work particularly well on the S&P.

Select your Time Frame

As you look at a chart and are evaluating market action type, it is important to consider the time frame of your chart. In fact, choosing which time frame is appropriate for you is almost as important as the type of market action and system you want to trade. You can take the same chart and time period and when you change the time frame, say from daily to weekly, the market action type may be completely different. We'll discuss three basic types of charts: *daily*, *weekly* and *intra-day*.

Daily Charts

The most common chart used by traders is the daily chart. Daily charts are the most common for several reasons. First, as most traders also have day jobs, they want to keep abreast of the market as much as possible without it intruding into their workday. The daily chart is perfect for this type of trader. You are able to review the markets each night and make your decisions for the next day.

Weekly Charts

Weekly charts are much more difficult to trade because it takes more discipline. To trade weekly charts you make your decisions on the weekends and don't make any changes until the next weekend. For most traders, this is very difficult to do. It's hard not to look at the market during the week and be tempted to move a stop loss or a money management stop, or not to want to keep your profits and exit the market early.

However, most people don't think of trading weekly charts, and a lot of traders agree that to make money in the markets you have to tread where the average traders do not tread. Weekly charts are one of those places.

Also, some systems work better on a weekly chart than on a daily. Very few people have the patience and the discipline to trade weekly charts. By their very nature, weekly charts smooth the price fluctuations of the daily chart. If there is a long trending market, we should be in the trend longer. We might get in the trend a little later than on the daily chart, and out later, but we will probably not get whipsawed as much in the directionless markets.

Intra-Day Charts

Intra-day charts are the 5-, 10-, 30-, and 60-minute charts that are compiled from intra-day tick data. Intra-day data, if used correctly, can give you a distinct advantage over using daily charts. If you have the time and energy, you can take advantage of the microscopic look at the markets.

However, to trade intra-day charts, you must give almost your full attention to the markets during the day. It is virtually impossible to have a full-time job and trade intra-day charts well. Relatively few traders are able to trade during the day. As with weekly charts, the relative lack of competition is probably in your favor trading intra-day.

Once you select the market action and therefore the system type you want to use and the time frame you will be trading, you're ready to start defining your trading rules.

Tips for Defining your Trading Rules

A large part of system development involves writing your rules clearly, without any ambiguity. In fact, you should be able to state your rules so clearly that anyone, using only the Data Window and a calculator, should be able to generate buy and sell signals. It sounds simple until you try it. Unless you have stated your rules in a way that is completely non-ambiguous, you won't be able to do it. And certainly, TradeStation won't be able to interpret your rules either.

When people first start developing systems, they tend to focus on the visual aspect. They try to describe their entry criteria, for example, based on the pattern they see in the chart. However, EasyLanguage is mathematical, so eliminating the visual aspect and focusing on the

mathematical aspect goes a long way to making sure your system does exactly what you want it to do.

Another stumbling block for beginning system developers is that sometimes they try to develop the entire system all at once. If you find an idea that you think is significant, don't rush to build a system around it. First, see if it's a valid idea. For example, you may want to write a ShowMe study to identify the criteria on the chart. Then, you can see if it's worth pursuing.

Once you decide an idea is worth pursuing and start writing a system, you still don't have to build the entire system. For instance, you can define your entry points only and use just a standard exit like a trailing stop. Or maybe work on the short side only, leaving out the long side. Many system developers have a handful of favorite exits which they use with the entries they develop. In other words, they spend their time developing the entries.

TradeStation Basics

Keep the following in mind when you are developing your systems:

- ✓ You can develop what are called reversal systems, which will reverse your position but never exit you from the market.

  To do this, you use Buy and Sell orders without ExitLong or ExitShort. This is not to say that you wouldn't get stopped out of the market, but your buy and sell orders do not ever exit you from the market. For example, if you are long 1 share, and the sell criteria is met, the system would sell 2 shares. If you are short 1 share, and the buy criteria is met, the system would buy 2 shares.

- ✓ When testing a system, make sure you take into account commissions, margin (if applicable) and slippage. Many times, you'll be evaluating a system and it will look real good until you factor in the above costs.

Most importantly, don't throw away an idea because it didn't work right away. Play with it, some ideas just need refining before you can see their true worth.

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 four trending systems that differ in their approach but are all designed to capture big moves and limit losses during directionless and/or volatile phases.

In This Chapter

Moving Average Crossover

Traditional Moving Average Crossover systems look for a crossing of the slow- and fast-moving averages in an effort to capture trends. They buy when the fast-moving average crosses over the slow-moving average, expecting the market to trend up, and sell when the opposite occurs, expecting the market to trend down. This technique can work well in trending market conditions, but when the market moves sideways or becomes choppy, the averages cross frequently and produce a lot of false signals that result in losing trades.

We wanted to develop a Moving Average Crossover (MACO) system that takes advantage of a trend but tries to avoid or minimizes any false signals. The goal was to identify trends but not act unless we could confirm that it was in fact the beginning of a trend.

To accomplish this, our system uses the crossing of the slow- and fast-moving averages to identify the possible start of a trend but then waits to generate a buy or sell signal until the up- or down-trend has been confirmed. It attempts to confirm the trend by placing buy or sell stop orders on the next bar, which remain active for a certain number of bars.

The system confirms an up-trend by identifying a “buy level.” Once the moving averages cross, the system sets the buy level to the highest high price of the last 12 bars plus 3 percent, and places an order to buy on the next bar if the buy level is reached (buy stop). The order remains active for 12 bars. In other words, if the buy level isn’t reached within 12 bars, the buy order is cancelled.

We designed EasyLanguage so that when you use the word BUY or SELL, TradeStation will reverse your position.

To exit a long or short position without reversing, you use the words EXITLONG or EXITSHORT, respectively. See the list of Reserved Words in the *EasyLanguage User's Manual*.

Likewise, the system confirms a down-trend by identifying a “sell level.” Once the moving averages cross, the system sets the sell level to the lowest low price of the last 12 bars minus 3 percent, and places an order to sell on the next bar if the sell level is reached (sell stop). Again, the order remains active for 12 bars, not indefinitely.

Of course, we will use inputs for the number of bars and percentages, this way we can test them and optimize them. What we are concerned with as we develop the system is a sound entry *concept*. Once we find a sound concept, we can decide exactly what numbers we will use.

We want the above buy and sell orders to reverse our position. In other words, if we are in a long position and a SELL order is filled, we are exited from a long position and entered into a short trade, and vice versa. This way, we are in the market and take advantage of a confirmed trend (at least the strong possibility of one).

Figure 2-1 shows an instance when a buy order was filled when the bar penetrated the buy level. Notice that the order was filled on the 8th bar after the crossover occurred. If the buy level had not been penetrated within 12 bars, the order would have been cancelled.

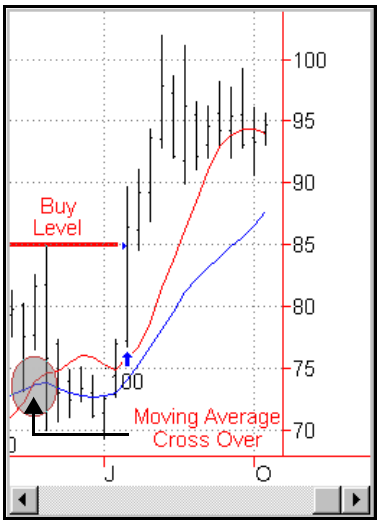


Figure 2-1. Buy signal generated upon confirmation of trend

This entry technique works well because it avoids your being in the market in drawdown periods immediately after a crossover occurs, as shown in Figure 1, and also avoids your entering the market on a false crossover signal.

Once we were happy with the entry criteria for our system, and found them to provide satisfactory results, we focused on the exits. Because it is a trending system, we wanted to exit when the trend was ending or starting to reverse.

In an attempt to anticipate the end or reversal of the trend, we will have our system exit from all long positions if the price reaches a substantially low price, or from all short positions if the price reaches a substantially high price, relatively speaking. We decided on a price level equal to or lower than the lowest low of the last 8 bars for the long exit, and equal to or higher that the highest high of the last 8 bars for the short exit. This works well to lock in profits. The higher the price goes, the higher our stop will be placed.

We are going to use the type of exit order that renders our position flat—in other words—it exits us from the market. We want to exit because we are confirming only that the trend is ending or potentially reversing, we already have criteria in place to enter the market when a strong trend is anticipated.

In evaluating the exit order, we realized that the above exit technique would sometimes result in a premature exit and cause the system to lose out on a large amount of profit. To compensate for this, we decided to include some re-entry criteria that will re-establish the original position if the trend continues.

To determine if the trend is continuing, we included the following rules: if a long position is terminated by this exit, the system monitors the price for 15 bars and if the price reaches the highest high of 10 bars, the system re-enters the long position. We can easily accomplish this by placing a buy stop at the highest high of the last 10 bars when the system exits.

We used the same criteria for the short side: if a short position is terminated by this exit and the price reaches the lowest low of 10 bars within 15 bars, the system re-enters the short position. Our goal is to make sure that the trend was in fact continuing. Figure 2-2 illustrates the re-entry technique.

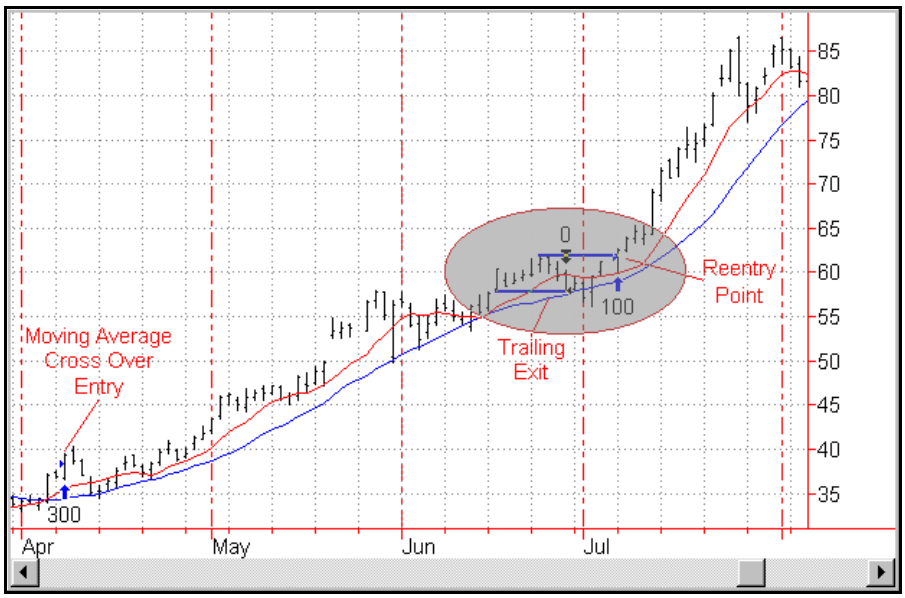


Figure 2-2. The re-entry criteria is met and a long order generated

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. Again, the long and short entries reverse your position, whereas the exits will close out your existing position and exit you from the market. The entries and exits are described next.

Long Entries

a) If the fast-moving average (9 bars) crosses over the slow-moving average (18 bars), find the highest high price of the last 8 bars and multiply it by 1.03. This is the long entry price.

b) Once a cross over occurs, place a buy stop order at the long entry price and keep the order active for the 12 bars after the cross over. Buy a specified number of shares, in this case 300.

c) If the system is in a long position and exited by any of the exit rules, place a buy stop order at the highest high price of the last 10 bars and keep the order active for 15 bars after the exit.

Short Entries

d) If the fast-moving average crosses under the slow-moving average, find the lowest low price of the last 8 bars and multiply it by .97. This is the short entry price.

e) Once the cross over occurs, place a sell stop order at the short entry price and keep the order active for the 12 bars after the cross over. Sell a specified number of shares, in this case 300.

f) If the system is in a short position and this position was covered (exited), place a sell stop order at the lowest low price of the last 10 bars and keep the order active for 15 bars after the exit.

Exits

g) Exit from any long position if the lowest low price of the last 8 bars is reached.

h) Exit from any short position if the highest high price of the last 8 bars is reached.

i) Place a money management protective stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts we’re trading .

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: Moving Average Crossover System

```
Input: FastLen(9), SlowLen(18), ChLen(12), TrailBar(8), Initial(300), ReBars(15), Reentry(100);
Vars: FastMA(0), SlowMA(0),LEntryPrice(0), SEntryPrice(0), LCount(-999), SCount(-999),
ReEntryCount(0), CurrentPosition(0);

FastMA = Average(Close,FastLen);
SlowMA = Average(Close,SlowLen);

{ Order Placement for Long Positions }
If FastMA crosses over SlowMA and BarNumber > 1 then Begin
        LEntryPrice = Highest(High,ChLen) * 1.03;
        LCount = BarNumber;
End;
If MarketPosition <> 1 and BarNumber < LCount + ChLen then
        Buy ("Cross Over Buy") Initial Shares next bar at LEntryPrice Stop;

{ Order Placement for Short Positions }
If FastMA crosses under SlowMA and BarNumber > 1 then Begin
        SEntryPrice = Lowest(Low,ChLen) *.97;
        SCount = BarNumber;
End;
If MarketPosition <> -1  and BarNumber < SCount + ChLen then
        Sell ("Cross Under Buy") Initial Shares next bar at SEntryPrice Stop;

{ Trailing Stop while in Position }
If MarketPosition = 1 then Begin
        LCount = -999;
        ExitLong ("LongTStop") next bar at Lowest(Low,TrailBar) Stop;
End;
If MarketPosition = -1 then Begin
        SCount = -999;
        ExitShort ("ShortTStop") next bar at Highest(High,TrailBar) Stop;
End;

{ Reentry Technique }
CurrentPosition = MarketPosition;

If CurrentPosition = 0 and CurrentPosition[1] = -1 then
        ReEntryCount = 1;
If CurrentPosition = 0 and CurrentPosition[1] = 1 then
        ReEntryCount = 1;

If MarketPosition = 0 and MarketPosition(1) = 1 and ReEntryCount < ReBars then Begin
        ReEntryCount = ReEntryCount + 1;
        Buy (“Long ReEntry”) ReEntry Shares next bar at Highest(High,10) Stop;
End;
If MarketPosition = 0 and MarketPosition(1) = -1 and ReEntryCount < ReBars then Begin
        ReEntryCount = ReEntryCount + 1;
        Sell (“Short ReEntry”) ReEntry Shares next bar at Lowest(Low,10) Stop;
End;
```

Inputs

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

Input	Default Value	Description
FastLen	9	The number of trailing bars used to calculate the fast moving average.
SlowLen	18	The number of trailing bars used to calculate the slow moving average.
ChLen	12	Number of bars during which the BUY or SELL order will remain active after the slow and fast moving averages have crossed.
TrailBar	8	Number of bars to use when finding the highest high price or lowest low price in order to place trailing stop orders.
Initial	300	Number of shares with which to open a new position after a moving average crossover and trend confirmation.
ReBars	15	Number of bars to consider when looking for the re-establishment of the initial trend in order to re-enter the market if taken out by any of the stop orders.
Reentry	100	Number of shares to use when re-entering the market after having a position closed by one of the stop orders.

In addition to the inputs, we declared several variables, as follows:

```
Vars: FastMA(0), SlowMA(0),LEntryPrice(0), SEntryPrice(0), LCount(-999), SCount(-999),
ReEntryCount(0), CurrentPosition(0);
```

Then, we used the Average function to calculate the fast- and slow-moving averages and assigned the values to the variables FastMA and SlowMA:

```
FastMA = Average(Close,FastLen);
SlowMA = Average(Close,SlowLen);
```

For a description of functions, variables and other EasyLanguage terminology, please refer to the *EasyLanguage User's Manual*.

The following sections discuss how the long and short entries, as well as the exits, are calculated.

Long Entries

When the fast moving average crosses over the slow moving average, the system calculates the value corresponding to 3% higher than the highest high of the last ChLen number of bars and stores it in the variable LEntryPrice. We also use the function BarNumber to obtain the bar number for the bar on which the cross over occurs and store it in the variable LCount. We will use LCount later to monitor the next TrailBar number of bars for confirmation of the trend:

```
If FastMA crosses over SlowMA and BarNumber > 1 then Begin
        LEntryPrice = Highest(High, ChLen) * 1.03;
        LCount = BarNumber;
End;
```

EasyLanguage Note: *BarNumber is a function that gives you the number of the specific TradeStation bar. Every bar on a chart is numbered in ascending order, starting from the left of the chart at 1. When checking for crossovers, you want to start checking on bar number 2 not bar number 1. This is because to check for cross overs, TradeStation looks at the mathematical values of the averages on the bar before and on the current bar. On bar 1, the previous values of the averages are 0 and therefore TradeStation will assume a cross over on bar 1 when none exists.*

The next set of instructions determines if the system is in a long position. We do so using the MarketPosition function, which returns a 1 when the system is long, a -1 when the system is short, and a 0 when it is flat. If the system is *not* in a long position, and no more than ChLen bars have elapsed since the cross over, the system will place a buy stop order to buy an Initial number of shares at LEntryPrice or anything higher. Remember, we calculated LEntryPrice in the prior set of instructions, at the point where the averages crossed, and we also stored the number of the bar in LCount:

```
If MarketPosition <> 1 and BarNumber < LCount + ChLen then
        Buy ("Cross Over Buy") Initial Shares next bar at LEntryPrice Stop;
```

If we were taken out of the market by one of the stop orders, and the market resumes the up-trend by breaking the highest high of the last 10 bars, we will re-enter the market. We will look for this to happen for ReBars number of bars after we are taken out of any long trade. We will use the variable ReEntryCount as a counter, to count the ReBars number of bars. The ReEntryCount variable will be set to 1 the bar after a long position is exited, and will increment by one thereafter. If the value of ReEntryCount is less than the input ReBars, then the buy order is placed:

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.

```
If CurrentPosition = 0 and CurrentPosition[1] = 1 then
        ReEntryCount = 1;

If MarketPosition = 0 and MarketPosition(1) = 1 and ReEntryCount < ReBars then Begin
        ReEntryCount = ReEntryCount + 1;
        Buy (“Long ReEntry”) ReEntry Shares next bar at Highest(High,10) Stop;
End;
```

We used the variable CurrentPosition to store the current market position and then to refer to the market position of one bar ago. You can use the function MarketPosition only to refer to the current market position or previous positions (up to 10 positions ago), you can’t identify the market position of a specific bar other than the current bar. However, what we can do is store the market position for each bar in the variable CurrentPosition and then refer to the value of the variable one bar ago.

Short Entries

When the fast-moving average crosses under the slow-moving average, the system calculates the lowest low price of the last ChLen bars minus 3% and stores this value in the variable SEntryPrice. We will also store the number of the bar at which the averages cross in the variable SCount. As with the variable LCount for the long entries, we will use SCount later to monitor the next TrailBar number of bars for confirmation of the trend:

```
If FastMA crosses under SlowMA and BarNumber > 1 then Begin
        SEntryPrice = Lowest(Low,ChLen) *.97;
        SCount = BarNumber;
End;
```

If the system is not in a short position, and no more than ChLen bars have elapsed since the cross under, the system will place a sell stop order to buy Initial number of shares at SEntryPrice or anything lower:

```
If MarketPosition <> -1  and BarNumber < SCount + ChLen then
        Sell ("Cross Under Buy") Initial Shares next bar at SEntryPrice Stop;
```

We will reenter the market if we were taken out by one of the stop orders and the market resumes the down-trend by breaking the lowest low of the last 10 bars. We will look for this to happen for ReBars number of bars after we are taken out of any long trade. To count these ReBars number of bars, we will use a counter—the variable ReEntryCount. This counter works the same on the short side as it does with the long side. To monitor market position, we use the function MarketPosition and the variable CurrentPosition:

```
If CurrentPosition = 0 and CurrentPosition[1] = -1 then
        ReEntryCount = 1;
```

```
If MarketPosition = 0 and MarketPosition(1) = -1 and ReEntryCount < ReBars then Begin
        ReEntryCount = ReEntryCount + 1;
        Sell ("Short ReEntry") ReEntry Shares next bar at Lowest(Low,10) Stop;
End;
```

Exit Orders

Once the system is in a long position, it will place a trailing stop at the lowest low of Trailbar bars:

```
If MarketPosition = 1 then Begin
        LCount = -999;
        ExitLong ("LongTStop") next bar at Lowest(Low,TrailBar) Stop;
End;
```

Notice that we set the variable LCount to -999. This keeps the system from placing another buy order after the cross over. For example, let's say the system generates a buy signal after a cross over and confirmation. Then on the 8th bar after the cross over an exit signal is generated by the trailing stop. At that point, LCount would still be less than 12 and the system could generate another buy signal. We want to limit the system to generating one buy signal per cross over.

The inverse technique is used for short trades:

```
If MarketPosition = -1 then Begin
        SCount = -999;
        ExitShort ("ShortTStop") next bar at Highest(High,TrailBar) Stop;
End;
```

Note: *As discussed in the next section, we set the system not to pyramid; however, the pyramid setting will not affect the above instructions. We have to use the variable to avoid another signal.*

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We did not include margin because we designed this system for stocks, and we did not include a default number of contracts to trade per order because we are specifying this using the input Initial.

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. In this system, this is specified by the input Initial. If the system did not specify how many shares/contracts to handle per transaction, this would be determined by the* ***Default Contracts*** *option on this tab.*

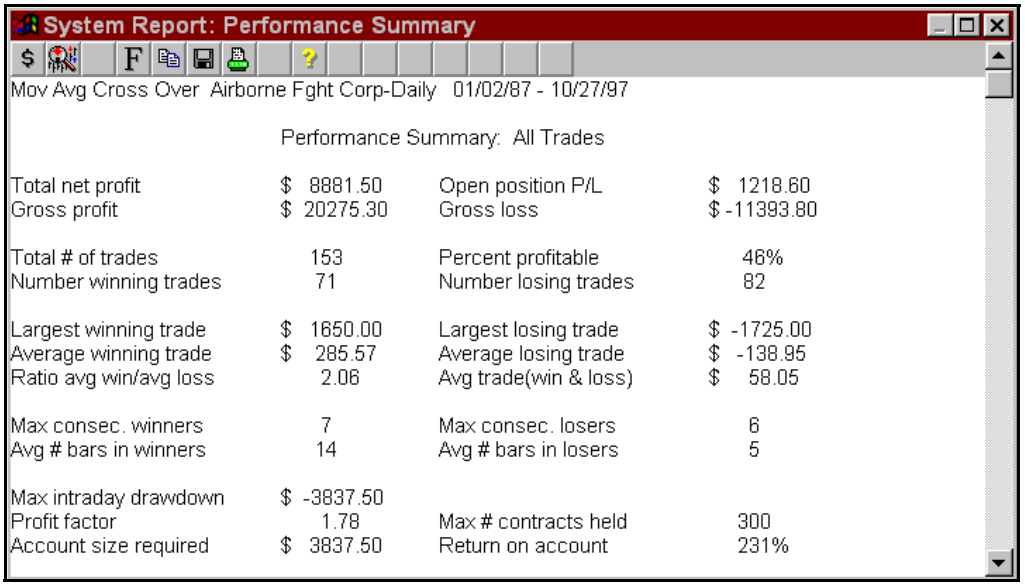
b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

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.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in 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

We applied the MACO system to a chart; the resulting System Report is shown in Figure 2-3.



System Report: Performance Summary

Mov Avg Cross Over Airborne Fght Corp-Daily 01/02/87 - 10/27/97

Performance Summary: All Trades

Total net profit	$ 8881.50	Open position P/L	$ 1218.60
Gross profit	$ 20275.30	Gross loss	$ -11393.80
Total # of trades	153	Percent profitable	46%
Number winning trades	71	Number losing trades	82
Largest winning trade	$ 1650.00	Largest losing trade	$ -1725.00
Average winning trade	$ 285.57	Average losing trade	$ -138.95
Ratio avg win/avg loss	2.06	Avg trade(win & loss)	$ 58.05
Max consec. winners	7	Max consec. losers	6
Avg # bars in winners	14	Avg # bars in losers	5
Max intraday drawdown	$ -3837.50		
Profit factor	1.78	Max # contracts held	300
Account size required	$ 3837.50	Return on account	231%

Figure 2-3. System Report for MACO system

Notice that the percentage of profitable trades is under 50 (46%), but the average wining trade is more than twice the average losing trade. Obviously, the higher the percent of profitable trades the better the system, but as we've discussed, trending systems by their nature have a large number of trades, a handful of which are highly profitable.

When evaluating your trending system, also look for big winning trades—the system should always be in the market whenever there is a sustained move in one direction.

You should always be concerned with the amount of data on which you back test your systems, and more so with the number of trades taken by the system throughout the tested data. Beware of trending systems that show few trades and the largest wining trade is close to the entire or a significant portion of the net profit.

There is no strict rule of thumb, but you should be happy with the net profits and with the average wining trade ***after excluding the largest wining trade***. Also, always look at the maximum intra-day drawdown. It should not be significant compared to the net profits, and of course you should be able to afford it.

Suggestions for Improvement

As we began testing our MACO system, we noticed that it sometimes missed a significant down- or up-trend (generally a down-trend). After closer inspection of these missed trends, we noticed that the trend was being initiated by a down day with an extremely large range, as illustrated in Figure 2-4.

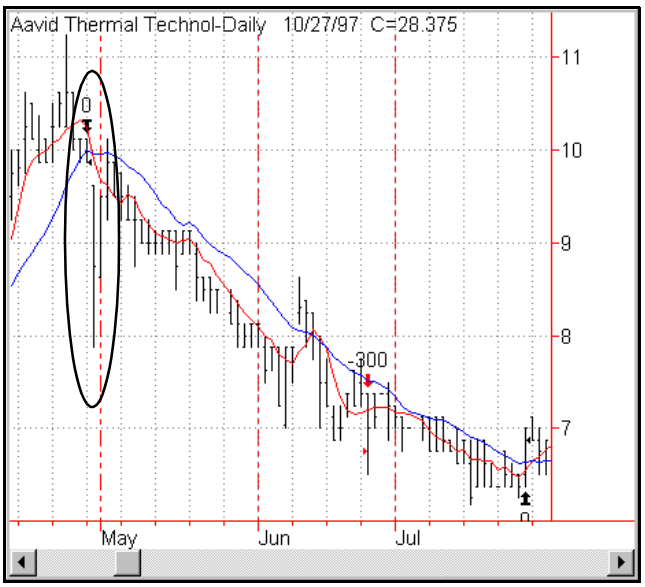


Figure 2-4. MACO system missing a potentially profitable signal

When the system looks for the lowest low of the last 10 bar, it includes this down bar and an order is not generated because the market was slowly down trending. To avoid this, when finding the sell price, we can find the lowest low price of ChLen number of bars, but offset it by one, so the current bar is not included:

```
If FastMA crosses under SlowMA and BarNumber > 1 then Begin
        SEntryPrice = Lowest( Low , ChLen )[1] *.97;
        SCount = BarNumber;
End;
```

And, when finding the buy price, find the highest high price of ChLen number of bars, and also offset it by one: HIGHEST(High , Chlen)**[1]**. Again, the purpose is not to include the day of the cross over, which usually is a bar with a greater than normal range, when finding buy and sell levels (this is the day that causes the averages to cross). When we make this minor modification to the long and short entries, the results of the system are improved. Figure 2-5 illustrates the same chart pattern, this time the system generated the more favorable signals.

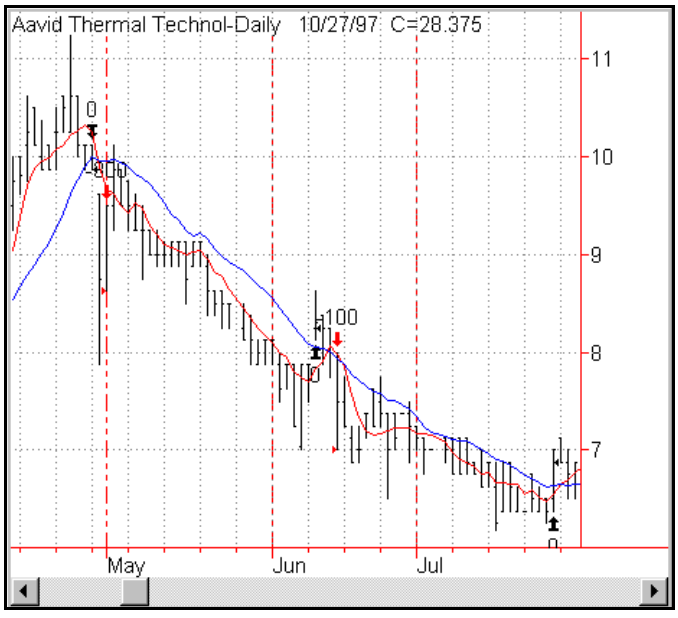


Figure 2-5. Improved MACO system applied — signal is generated

Also, this system is somewhat active. To improve the performance, we may want to look for an additional filter to avoid trades when the market is moving sideways and is very volatile (high bar ranges).

One way we could do this is with directional indicators (like the ADX or the DMI) in combination with volatility measurements of the market. For example, when the directional indicators are indicating no trend and volatility is high, we might want to stay away from entry orders or move entry levels further away from the current market price.

Moving Average Convergence Divergence (MACD) Breakout

As we discussed, for the most part, markets tend to be choppy or move sideways, falling into up- or down-trends only about 15% of the time. Like the Moving Average Crossover system, this system, the Moving Average Convergence Divergence (MACD) Breakout attempts to determine when the market is starting to trend in order to take advantage of the trend.

The MACD Indicator was developed by Gerald Appel, and is both a trend following indicator and an oscillator. It calculates moving averages that can monitor and signal trends, and it also helps spot divergences that aid in determining when a trend is overbought or oversold.

However, instead of using fast- and slow-moving average cross overs to identify the possible beginning of a trend, this system creates a channel using a differential calculation of the Moving Average Convergence Divergence (MACD) and an exponential average of the MACD. Once a channel is established, the system looks for a break out of the upper band to establish a long position and a break out of the lower band to establish a short position.

To create the channel, the system monitors the MACD differential to determine when it crosses under or over zero (0). The system calculates the differential as follows:

```
MVal = MACD(Close,FastMA,SlowMA);
MAvg = XAverage(MACD(Close,FastMA,SlowMA),MacdMA);
MDif = MVal - MAvg;
```

The first line calculates the MACD using the close price and two lengths (in this system we use 12 and 26). The second line calculates the exponential moving average of the MACD (in this system we use a length of 9). The third line subtracts the exponential average of the MACD form the MACD, resulting in the differential value we compare to zero (0).

As soon as 4 cross overs or cross unders are found, sell stop and buy stop orders are placed. These orders stay active as long as there are 4 cross overs and/or unders within the last 50 bars.

Our idea is that when the MACD differential crosses under or over zero frequently, we consider the market to be in a sideways mode. This system looks at the last 50 bars and when it finds 4 cross overs and/or cross unders within the 50 bars, it uses the bars making up the cross overs and/or unders and establishes the channel.

Where range is the difference between the high and low prices of a bar, true range takes into account the close of the previous bar and represents the actual range of movement from one bar to the next.

The upper band, or buy level, is determined by finding the highest high of the last 4 cross overs or unders, and adding an extra measure. The extra measure is half of the 4-bar average of the true range. The lower band, or sell level, is determined in a similar way, finding the lowest low of the same 4 bars, and subtracting one half the 4-bar average of thetrue range. When the system finds the 4 cross overs and/or unders, it places buy and sell stops at the buy and sell levels, respectively. Then, when the price penetrates the buy level, a buy order is generated. Likewise, when the price penetrates the sell level, a sell order is generated.

Sometimes, we find it useful to create an indicator, ShowMe or PaintBar study to help visualize our idea. In fact, a lot of times we'll create an indicator or study just to see (visually) if an entry idea has any merit before we consider creating a system with it. In this instance, we created an indicator with three plots, called MACD Breakout. The first plot paints a dot on the bars that cross under or over, and the other two plots draw the upper and lower bands.

Figure 2-6 shows the system and the indicator we created applied to a chart. We also applied the standard MACD indicator (making sure to use the same inputs for both indicators). Notice that the system found 4 cross overs and/or unders within 50 bars and therefore placed buy stop and sell stop orders using the buy and sell levels. When the buy level was penetrated, the order was generated. The indicator shows the channel and cross over and/or under bars.

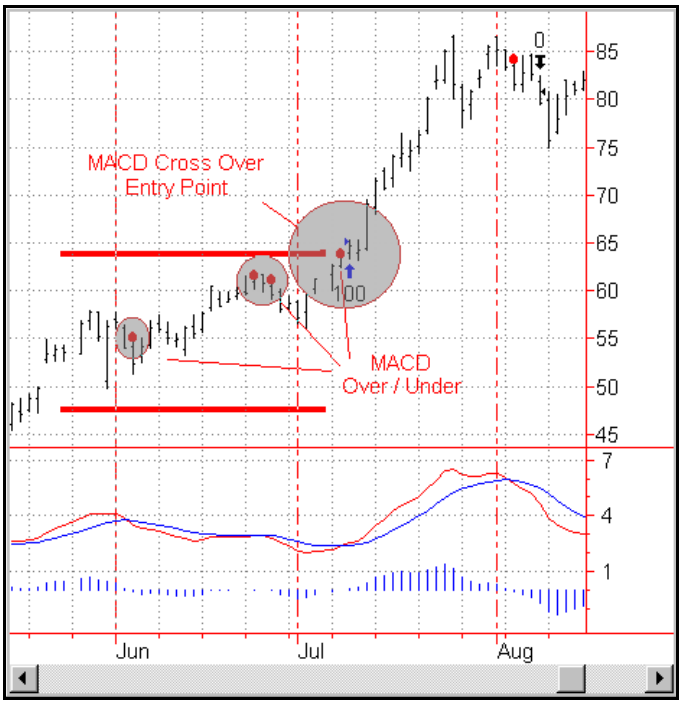


Figure 2-6. MACD Breakout system and indicator applied to the chart

Once we decided we had valid entries, we looked at how to exit. Since it is a trending system, we decided to use trailing stops. We will exit out of our long positions when the market drops to the lowest low of the last 6 bars, and we will exit out of our short positions at the highest high of the last 6 bars. Notice the bar on which the system generated the exit signal in Figure 2-6 above. Again, we used inputs for all the numbers so that we can adjust the system at will and, of course, optimize the system.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some set up work, which involved calculating the MACD differential and storing the values for later retrieval. Our setup, entries and exits are as follows:

Setup

a) Calculate the value of the MACD, using the MACD function.

b) Calculate the exponential moving average of the MACD.

c) Calculate the difference between the first two values—this we will call MACD differential.

d) Find the last 4 instances where the MACD differential crossed under or over zero (0).

Long Entries

e) If the last 4 crosses found in Step d) occurred in the last 50 bars, place a buy stop order at the highest high of the cross over/under bars plus half the average true range of the last 4 bars.

Short Entries

f) If the last 4 crosses found in Step d) occurred in the last 50 bars, place a sell stop order at the lowest low of the cross over/under bars plus half the average true range of the last 4 bars.

Exits

g) Exit from any long position if the price drops to the lowest low of the last 6 bars.

h) Exit from any short position if the price reaches the highest high of the last 6 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: Moving Average Convergence Divergence Breakout System

```
Input: FastMA(12),SlowMA(26),MacdMA(9), NCOs(4), NBars(50), Trailbar(6);
Vars: MVal(0), MAvg(0), MDif(0), Counter(0), TotalBars(0), HighestHi(0), LowestLo(0);
Array: CO[2,50](0);

MVal = MACD(Close,FastMA,SlowMA);
MAvg = XAverage(MACD(Close,FastMA,SlowMA),MacdMA);
MDif = MVal - MAvg;

{ Store the MACD Cross information in the array CO }
If  MDif Crosses Over  0  OR  MDIF Crosses Under  0 Then Begin
        For counter = 0 to 49 Begin
                CO[ 0 , 50 - Counter ] = CO[ 0 , 49 - Counter ];
                CO[ 1 , 50 - Counter ] = CO[ 1 , 49 - Counter ];
                CO[ 2 , 50 - Counter ] = CO[ 2 , 49 - Counter ];
        End;
        CO[ 0 , 0 ] = BarNumber;
        CO[ 1, 0 ] = High;
        CO[ 2, 0 ] = Low;
End;

{ Find the Highest High and the Lowest Low of the cross over and under bars }
HighestHi = -1;
LowestLo = 9999;
For counter = 0 to NCOs-1 Begin
        If CO[ 1 , counter ] > HighestHi then
                HighestHi = CO[ 1 , counter ];
        If CO[ 2 , counter ] < LowestLo then
                LowestLo = CO[ 2 , counter ];
End;

{ If the last x number of  crosses occured in the last 50 bars, then place the buy and sell orders }
Totalbars = BarNumber - CO[ 0 , NCOs - 1];
If TotalBars < NBars then Begin
        Buy at HighestHi + Average(TrueRange,4)*.5 Stop;
        Sell at LowestLo - Average(TrueRange,4)*.5 Stop;
End;

{ Trailing stop orders }
ExitLong next bar at Lowest( Low , TrailBar ) Stop;
ExitShort next bar at Highest( High , TrailBar ) Stop;
```

Inputs

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

Input	Default Value	Description
FastMA	12	The number of trailing bars used to calculate the fast moving average of the MACD.
SlowMA	26	The number of trailing bars used to calculate the slow moving average of the MACD.
MacdMA	12	Number of bars used in the calculation of the exponential moving average of the MACD.
NCOs	4	Number of MACD differential cross overs and/or unders that will be used to build the channel.
NBars	15	Maximum number of bars in which the cross overs and/or unders must occur in order for the system to generate the buy and sell stop orders.
TrailBar	6	Number of bars to consider when placing a trailing stop order at the lowest low and highest high, for long and short orders respectively.

In addition to the inputs, we declared several variables, as follows:

```
Vars: MVal(0), MAvg(0), MDif(0), Counter(0), TotalBars(0), HighestHi(0), LowestLo(0);
```

Then, we used the MACD and XAverage functions to calculate the MACD and the exponential average of the MACD, and assigned the values to the variables MVal and MAvg and MDif:

```
MVal = MACD(Close,FastMA,SlowMA);
MAvg = XAverage(MACD(Close,FastMA,SlowMA),MacdMA);
MDif = MVal - MAvg;
```

The following sections discuss the use of arrays as well as how the long and short entries and the exits are calculated.

Arrays

This system used the information on the four most recent MACD cross over and/or unders to build a channel. There are two ways in which you can reference this information. You can, on each bar, look back 50 bars and re-calculate the information. This is cumbersome and time consuming. The second way, using an array to store the information on a bar-by-bar basis as you encounter the cross over and/or unders, is much simpler and efficient.

Arrays are actually variables—they store values. However, they are special variables because rather than being a place holder for one value, they are like spreadsheets and can hold from a few to hundreds of pieces of information under one name. Values stored in an array are referred to as *elements*. You can think of an array as being like a spreadsheet. For example, we want to store the bar number for the bars on which the MACD cross occurred, as well as the high price and low price. To do so, we define an array that has 3 columns and 51 rows, for a total of 153 elements. This way, we can store the information for the last 50 bars on which the MACD crossed over or under zero (0) for quick easy access at any time. The Array Declaration statement is:

```
Array: CO[2,50](0);
```

The array is called CO and it has 3 columns (0, 1 and 2) and 51 rows (0 - 50). To store and later reference values in an array, you refer to the column and row. For example, we can store a value

in CO[0,1]. To later retrieve this element, we reference the same array name and column and row numbers.

The most recent cross will be stored in row 0 of the array and the oldest in row 51. Every time that the MACD differential (MDif) crosses zero, the array will discard the information of the oldest cross, and move the information of each cross one row up. When this is done, the new MACD cross will be stored in row 0:

To fill arrays, you use a mechanism called **loops**. It is a simple technique that lets you run through a set of instructions a set number of times, say for 9 times, or while some condition is true.

Notice the For loop used to fill the CO array. When TradeStation reaches this part of the analysis technique, it performs the instructions inside the Block IF-THEN statement 50 times, with the variable Counter automatically incrementing by one each time it performs the instructions.

```
{ Store the MACD Cross information in the CO Array }
If  MDif Crosses Over  0  OR  MDIF Crosses Under  0 Then Begin
        For Counter = 0 to 49 Begin
                CO[ 0 , 50 - Counter ] = CO[ 0 , 49 - Counter ];
                CO[ 1 , 50 - Counter ] = CO[ 1 , 49 - Counter ];
                CO[ 2 , 50 - Counter ] = CO[ 2 , 49 - Counter ];
        End;
        CO[ 0 , 0 ] = BarNumber;
        CO[ 1, 0 ] = High;
        CO[ 2, 0 ] = Low;
End;
```

Storing the cross over and/or under information makes it very easy to find the highest high and lowest low among the bars in the array. Since the number of crosses being looked at by the system to define the channel is 4 (specified using the input NCOs), the system will find the highest high and the lowest low among the most recent 4 crosses:

```
{ Find the Highest High and the Lowest Low of the cross over and under bars }
HighestHi = -1;
LowestLo = 9999;
For Counter = 0 to NCOs-1 Begin
        If CO[ 1 , Counter ] > HighestHi then
                HighestHi = CO[ 1 , counter ];
        If CO[ 2 , Counter ] < LowestLo then
                LowestLo = CO[ 2 , counter ];
End;
```

Entry Orders

If the last 4 crosses occurred in the last 50 bars (or however many bars are specified using the input NBars), the system will place a buy stop order at the highest high of the cross over/under bars plus half the average true range of the last 4 bars. It will also place a sell stop order at the lowest low minus half the average true range of the last 4 bars:

```
{ If the last x number of  crosses (default 4) occured in the last 50 bars,
then place the buy and sell orders }
Totalbars = BarNumber - CO[ 0 , NCOs - 1];
If TotalBars < NBars then Begin
        Buy at HighestHi + Average(TrueRange,4)*.5 stop;
        Sell at LowestLo - Average(TrueRange,4)*.5 stop;
End;
```

Exits

The system will place a trailing stop at the lowest low of 6 bars for long positions, and highest high of the last 6 bars for short positions.

```
{ Trailing stop orders }
ExitLong next bar at Lowest( Low , TrailBar ) Stop;
ExitShort next bar at Highest( High , TrailBar ) Stop;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for the default number of contracts to trade, commission and slippage. We did not include margin because we designed this system for stocks.

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. Unless specified within the system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in 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 2-7 shows a sample System Report for the MACD Breakout system. As expected, it is a classic trending system report. The percentage of wining trades is under 50 and the ratio of winning versus losing trades is close to 2. These are the 2 most relevant characteristics of trending systems. We also realized though, that the largest wining trade is about a fourth of the total net profit, this is a potential red flag—something worth considering and investigating on this chart.

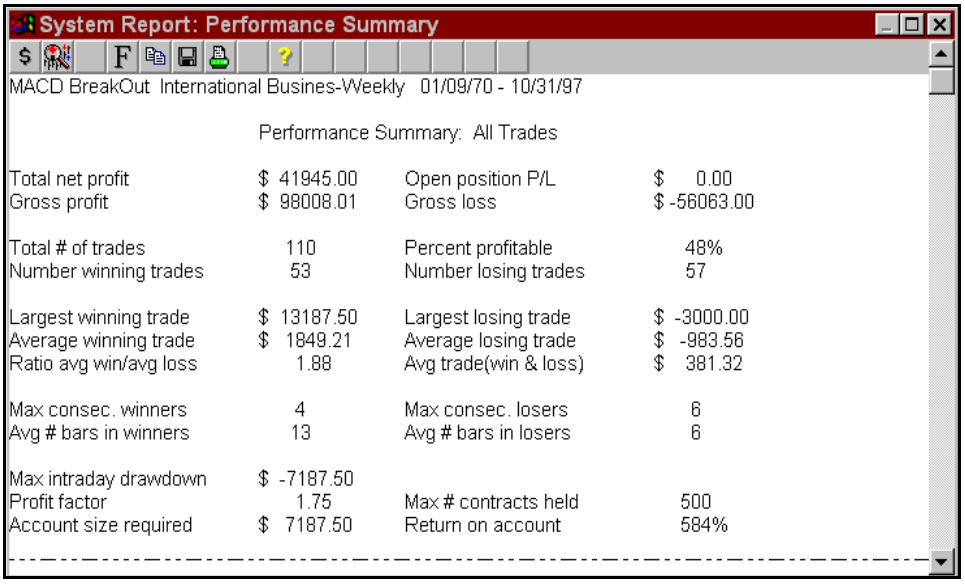


Figure 2-7. System Report for MACD Breakout system

Figure 2-8 shows the performance summary of all short trades. The percent of winning trades remains the same but the average winning trade versus losing trade ratio is down to 1.41. There is also a larger drawdown when looking at the performance of the short trades versus the system in general.

Figure 2-8. System Report showing performance summary of all short trades

This may suggest that this system is using a valid entry point (it kept the 48% of winning trades) but it is flawed when covering the short side. There are several factors we can consider, for example, the market bias. If the market has been in a overall up-trend, then shorting the instrument would not result in big earnings. We could also change the short side of the system so it uses another technique or disregards it altogether. Although we will lose out on opportunities in bearish trends, the system does not need to go short if the short trades hurt the overall performance.

Suggestions for Improvement

Figure 2-9 shows a chart of this system on a stock that is is trending rather than moving sideways.

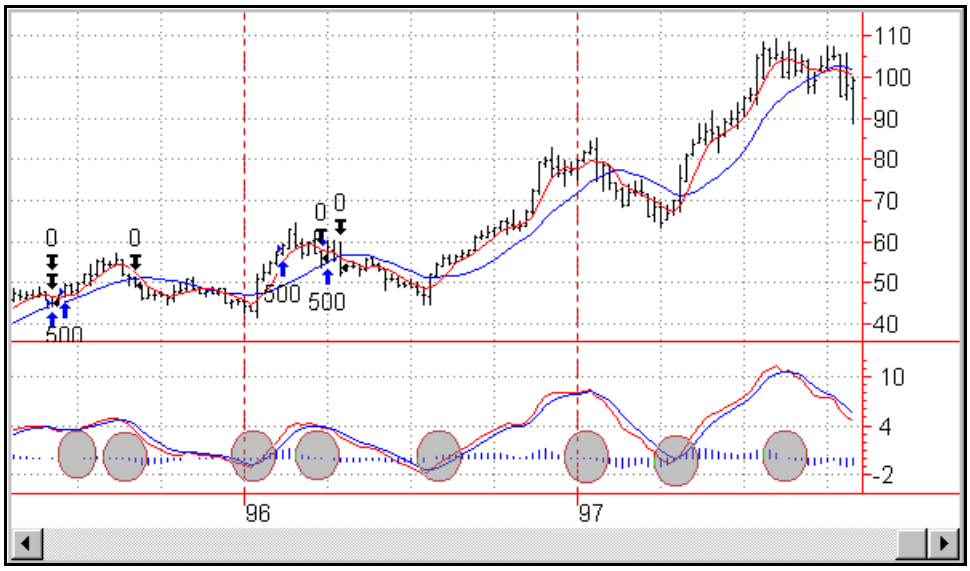


Figure 2-9. The MACD Breakout system applied on a trending market that has no sideways movement.

As you can see, after mid-1996, it went into an up-trend, in January 1997, it initiated a smaller down-trend, and then in March 1997, it started a new up-trend without ever moving sideways. However, this system is designed for markets that primarily move sideways and attempts to anticipate a break out of the sideways channel.

To make the system more sensitive and react better to symbols like the one in Figure 2-9, which spend little time moving sideways, we can lower the input named COs, which specifies how many cross overs and/or unders have to occur within the specified number of bars in order to create the channel. Figure 2-10 shows the system applied to the same symbol, with the input CO changed from the default of 4 to 2.

Figure 2-10. The MACD Breakout system applied to the same chart, with the input CO changed from 4 to 2

Lowering the value of the input does result in the desired effect; however, one drawback we noticed is that it makes the system prone to giving false signals if the market becomes more volatile. This type of fine-tuning should be done once you determine the market on which you are testing a system.

Trend Line Automatic

One of the oldest methods of analyzing trending markets is by using trend lines. Both TradeStation 4 and SuperCharts 4 provide the Trend Line - Automatic Indicator, which finds up- and down-trends in the market and automatically draws the corresponding trend lines. We wanted to create a system based on the indicator; our thinking was that we could take advantage of the EasyLanguage techniques used by the indicator to identify trends as well as buy and sell levels—we would want to buy or sell when these levels were penetrated.

Like the indicator, we designed the system to keep track of the last 10 *swing lows*. If a higher swing low is found (a swing low that is higher than the previous swing low), the market is considered to be in a up-trend, and a trend line is drawn. Likewise, the system will keep track of the last 10 *swing highs*. When the most recent swing high found is lower than the 9 previous swing highs, the market is considered to be in a down-trend and the trend line is drawn.

A swing low is defined as a bar that has a low price that is equal to or lower than *x* number of preceding bars and a lower low than the next *x* number of bars. A swing high is defined as a bar that has a high that is equal to or higher than that of *x* number of preceding bars and a higher high than the next *x* number of bars. The theory is that the swing low or swing high is the point at which price movement changed direction, or a price reversal occurred.

Once we draw the trend lines, we'll have the system monitor the price data for a penetration of the trend lines. When the price crosses either the up- or down-trend line, we'll have the system place a long or short entry order on the next bar at the open price, respectively. Figure 2-11 shows the system applied to a chart.

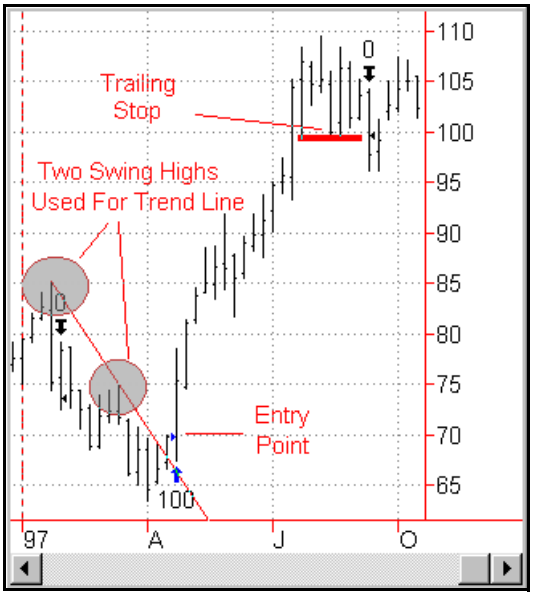


Figure 2-11. Trend Line Automatic system applied to a chart

After analyzing the trades generated by the system, we realized that when we use the swing high and swing low method to draw trend lines, some are drawn with virtually zero slope, and a penetration of these horizontal lines doesn't necessarily indicate a significant break out. Therefore, our system generates a lot of false signals. To compensate for this, we decided to take into account only trend lines that have a slope greater than 10 degrees. When we modified the system in this way, it ended up with much more reliable entry results.

Once we had more reliable entry results and a system we think we can work with, we decided to use a trailing stop to exit out of these trades. We will exit out of our long positions when the market touches the lowest low of the last 6 bars, and we will exit out of our short positions when the price reaches the highest high of the last 6 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 set up work, which involved calculating and drawing the trend lines. Our setup, entries and exits are as follows:

Setup

a) Find the last 10 swing high bars. We define a swing high bar as a bar that has a high that is equal to or higher than the preceding 4 bars as well as a higher high than the next 4 bars.

b) Find the last 10 swing low bars. We define a swing low bar as a bar that has a low that is equal to or lower than the preceding 4 bars as well as a lower low than the next 4 bars.

c) If the most recent swing high found in Step a) is lower than any of the 9 preceding swing highs, draw a trend line uniting these swing highs and extend the trend line to the right. This is the down-trend line.

d) If the most recent swing low found in Step b) is higher than any of the 9 preceding swing lows, draw a trend line uniting these swing lows and extend the trend line to the right. This is the up-trend line.

Long Entries

e) If the close of the current bar crosses over the down-trend line and this trend line has a slope greater than 10 degrees, then buy on the next bar at the open.

Short Entries

f) If the close of the current bar crosses over the up-trend line and this trend line has a slope greater than 10 degrees, then sell on the next bar at the open.

Exits

g) Exit from any long position at the lowest low of the last 6 bars.

h) Exit from any short position at the highest high of the last 6 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: Trend Line Automatic System

```
Inputs: RHStren(4), RLStren(4), Trailbar(6);

Vars: RHTLRef(-1), RHTLRef2(-1), RHSet(-1), RHSet2(-1), RHArrayVal(0), RHColorVar(0),
RLColorVar(0), BarsPast(0), RLTLRef(-1), RLTLRef2(-1), RLSet(-1), RLSet2(-1), RLArrayVal(0),
BuyPrice(0), PrevBuyPrice(0), SellPrice(0), PrevSellPrice(0);

Array: RHDate[10](0), RHTime[10](0), RHVal[10](0), RLDate[10](99999), RLTime[10](99999),
RLVal[10](99999);

{ Find new Swing High bars and draw corresponding trend lines }
If SwingHighBar(1, High, RHStren, RHStren+1)=RHStren then Begin
        For Value1=9 downto 0 Begin
                RHDate[Value1+1]=RHDate[Value1];
                RHTime[Value1+1]=RHTime[Value1];
                RHVal[Value1+1]=RHVal[Value1];
        End;
        RHDate[0]=Date[RHStren];
        RHTime[0]=Time[RHStren];
        RHVal[0]=High[RHStren];
        For Value22=1 to 10 Begin
                If RHVal[Value22]>RHVal[0] then Begin
                        RHArrayVal=Value22;
                        Value22=11;
                End;
        End;
        If Value22<>11 then Begin
                If RHSet>=0 then Begin
                        RHSet2=TL_SetExtRight(RHTLRef, False);
                        TL_Delete(RHTLRef);
                End;
                RHTLRef=TL_New(RHDate[RHArrayVal], RHTime[RHArrayVal], RHVal[RHArrayVal],
                         RHDate[0], RHTime[0], RHVal[0]);
                RHSet=TL_SetExtRight(RHTLRef, True);
        End;
End;
```

```
{ Find new Swing Low bars and draw corresponding trend lines }
If SwingLowBar(1, Low, RLStren, RLStren+1)=RLStren then Begin
        For Value1=9 downto 0 Begin
                RLDate[Value1+1]=RLDate[Value1];
                RLTime[Value1+1]=RLTime[Value1];
                RLVal[Value1+1]=RLVal[Value1];
        End;
        RLDate[0]=Date[RLStren];
        RLTime[0]=Time[RLStren];
        RLVal[0]=Low[RLStren];
        For Value22=1 to 10 Begin
                If RLVal[Value22]<RLVal[0] then Begin
                        RLArrayVal=Value22;
                        Value22=11;
                End;
        End;
        If Value22<>11 then Begin
                If RLSet>=0 then Begin
                        RLTLRef2=TL_SetEnd(RLTLRef, Date,Time,TL_GetValue(RLTLRef, date,
                                time));
                        RLSet2=TL_SetExtRight(RLTLRef, False);
                        TL_Delete(RLTLRef);
                End;
                RLTLRef=TL_New(RLDate[RLArrayVal], RLTime[RLArrayVal], RLVal[RLArrayVal],
                        RLDate[0], RLTime[0], RLVal[0]);
                RLSet=TL_SetExtRight(RLTLRef, True);
        End;
End;

{ Place orders to enter long positions }
If RHTLRef <> -1 then Begin
        BuyPrice = TL_GetValue( RHTLRef , Date, Time );
        PrevBuyPrice = TL_GetValue( RHTLRef , Date[1] , Time[1] );
        If Close Crosses Over BuyPrice AND Arctangent( PrevBuyPrice / BuyPrice)>10 then
                Buy next bar at Market;
End;

{ Place orders to enter short positions }
If RLTLRef <> -1 then Begin
        SellPrice = TL_GetValue( RLTLRef , Date , Time );
        PrevSellPrice = TL_Getvalue( RLTLRef , Date[1] , Time[1] );
        If Close crosses under SellPrice AND Arctangent(SellPrice / PrevSellPrice)>10 then
                Sell next bar at Market;
End;

{ Trailing protective stop orders }
ExitLong next bar at Lowest( Low , TrailBar ) Stop;
ExitShort next bar at Highest( High , TrailBar ) Stop;
```

Inputs

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

Input	Default Value	Description
RHStren	4	The number of bars to use as the strength of the swing high bar pattern.
RLStren	15	The number of bars to use as the strength of the swing low bar pattern.
TrailBar	6	The number of bars to consider when placing a trailing stop order at the lowest low and the highest high of long and short orders, respectively.

In addition to the inputs, we declared several variables, as follows:

```
Vars: RHTLRef(-1), RHTLRef2(-1), RHSet(-1), RHSet2(-1), RHArrayVal(0), RHColorVar(0),
RLColorVar(0), BarsPast(0), RLTLRef(-1), RLTLRef2(-1), RLSet(-1), RLSet2(-1),
RLArrayVal(0), BuyPrice(0), PrevBuyPrice(0), SellPrice(0), PrevSellPrice(0);
```

The following sections discuss the use of arrays as well as how the long and short entries and the exits are calculated.

Arrays

This system uses arrays to store the last 10 swing highs and swing lows. We will use 3 arrays for the swing highs and 3 arrays for the swing lows.

We used six one dimensional arrays; you can just as easily use multi-dimensional arrays, like we did with the MACD Breakout System; the end result is the same.

The swing high arrays use RH as the prefix, and the swing low arrays use RL as the prefix. These arrays will store the date, time and price data for the swings. The Array Declaration statement is:

```
Array: RHDate[10](0), RHTime[10](0), RHVal[10](0), RLDate[10](99999),
RLTime[10](99999), RLVal[10](99999);
```

The most recent swing, low or high, will be stored in row 0 of the corresponding array and the oldest in row 10. Every time that a new swing is found, the array will discard the information of the oldest swing, and move the information of each swing one row up. When this is done, the information for the new swing is stored in row 0:

```
If SwingHighBar(1, High, RHStren, RHStren+1)=RHStren then Begin
        For Value1=9 downto 0 Begin
                RHDate[Value1+1]=RHDate[Value1];
                RHTime[Value1+1]=RHTime[Value1];
                RHVal[Value1+1]=RHVal[Value1];
        End;
RHDate[0]=Date[RHStren];
RHTime[0]=Time[RHStren];
RHVal[0]=High[RHStren];
End;
```

The system will then look at the last swing high, and compare it to each one of the previous 9 swings. If the most recent swing high is lower than any of the other 9, the old trend line is deleted, and a new trend line is drawn using the new swing high:

```
For Value22=1 to 10 Begin
        If RHVal[Value22]>RHVal[0] then Begin
                RHArrayVal=Value22;
                Value22=11;
        End;
End;
```

For information on the functions used to draw trend lines, please refer to the PowerEditor on-line help. They all have the prefix TL.

When you draw trend lines, you must use a numeric variable to hold the ID number for the trend line. In our system, we use RHTLRef.

```
If Value22<>11 then Begin
        If RHSet >= 0 then Begin
                RHSet2=TL_SetExtRight(RHTLRef, False);
                TL_Delete(RHTLRef);
        End;
        RHTLRef=TL_New(RHDate[RHArrayVal], RHTime[RHArrayVal], RHVal[RHArrayVal],
                RHDate[0], RHTime[0], RHVal[0]);
        RHSet=TL_SetExtRight(RHTLRef, True);
End;
```

The system will then perform the same procedure with the swing lows using the RLDate, RLTime and RLVal arrays.

Long Entries

If the close of the current bar crosses over the down-trend line and this trend line has a slope greater than 10 degrees, then the system will place a buy order on the next bar at the open. RHTLRef is a variable that holds the ID number of the down-trend line. We set the value of this variable to -1, and it will hold this value until the first trend line is drawn. Therefore, before we obtain the trend line information and check for the cross over and slope, we will check RHTLRef to make sure it is not equal to -1:

```
If RHTLRef <> -1 then Begin
        BuyPrice = TL_GetValue( RHTLRef , Date, Time );
        PrevBuyPrice = TL_GetValue( RHTLRef , Date[1] , Time[1] );
        If Close Crosses Over BuyPrice AND ArcTangent( PrevBuyPrice / BuyPrice) >10 then
                Buy next bar at Market;
End;
```

Short Entries

If the close of the current bar crosses under the up-trend line and this trend line has a slope greater than 10 degrees, then the system will place a sell order on the next bar at the open. As with the long entry, we use the variable RLTLRef to make sure a trend line is drawn before we obtain the trend line information and check for both the cross over and slope:

```
If RLTLRef <> -1 then Begin
        SellPrice = TL_GetValue( RLTLRef , Date , Time );
        PrevSellPrice = TL_Getvalue( RLTLRef , Date[1] , Time[1] );
        If Close crosses under SellPrice AND Arctangent(SellPrice / PrevSellPrice)>10 then
                Sell next bar at Market;
End;
```

Exits

The system places a trailing stop at the lowest low of 6 bars for long positions and the highest high of 6 bars for short positions.

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

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for the default number of contracts to trade, commission and slippage. We did not include margin because we designed this system for stocks.

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. Unless specified within the system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in 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 2-12 shows the System Report for the Trend Line Automatic System.

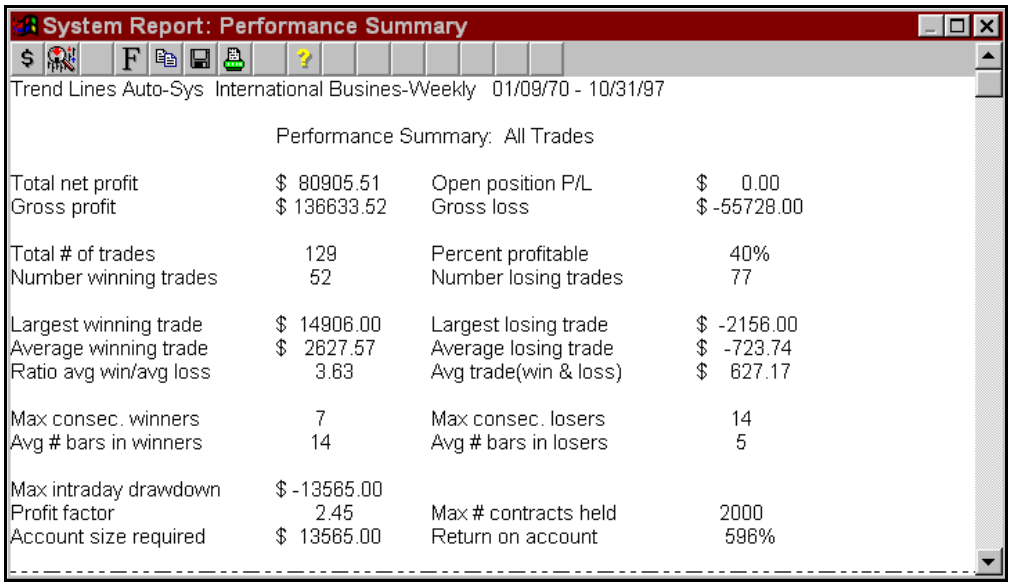


Figure 2-12. System Report for Trend Line Automatic system

The System Report is typical for a trending system: the percent of winning versus losing trades is 40%, and it has an outstanding average winning versus average losing trade ratio at 3.63. It has a large winning trade of $14,906 yet even when we take it out of our net profits of $80,905, we're left with $65,999, which is still more than acceptable to us. One item to notice is the high number of consecutive losers, this is one of the characteristics of trending systems and a main reason they are psychologically difficult to trade.

Suggestions for Improvement

As we tested the system, we noticed that it had a certain tendency to be stopped out from a position because of a minor correction in the trend and miss the rest of the move. What we may want to do is try adding a re-entry technique similar to the one explained in the Moving Average Crossover system.

Also, the input RHStrength and RLStrength determines the number of bars used to find a swing high or swing low bar, respectively, when drawing trend lines. In order to make the system more sensitive to up-trends, we may lower the value of RLStrength. Likewise, in order to make the system more sensitive to down-trends, we can lower the value of RHStrength. However, as we make the system more sensitive, it will be easier to be "fooled" by noise in the market, which will result in more frequent false signals.

Trend Follower

Like all trending systems, this system attempts to take advantage of a trend in the market, either up or down. To anticipate a trend, this system uses several factors:

- It establishes a top and bottom band using the open price and a volatility multiplier.
- It uses a 3-line moving average to determine market mode, bearish or bullish.
- It calculates an exponential moving average to compare to the bands.

We defined the top band as the open of the first available bar *plus* the volatility of 8 bars ago times a factor, and the bottom band is defined as the open of the first available bar *minus* the volatility of the last 8 bars times the same factor. We used volatility to calculate the bands because we wanted to use a value that, when crossed, represented a significant movement in price. We decided to multiply the volatility times a factor so that we could control the width of the bands and increase or decrease sensitivity at will.

To determine the current trend or market mode, we use three moving averages of different lengths (4, 9 and 18 bars). When the fast moving average is greater than the medium moving average and the medium moving average is greater than the slow moving average, we consider the market to be bullish, or in an up-trend. Conversely, when the fast moving average is less than the medium moving average and the medium moving average is lower than the slow moving average, we consider the market to be bearish, or in a down-trend.

We also calculate a 4-bar exponential moving average. We will initiate a long position when we consider the market to be bullish and the exponential average moves over the top band. Also, when this occurs, the original top band becomes the bottom band and a new top band is defined. The opposite will be true for the short side.

We decided to compare the exponential moving average to the top and bottom bands rather than a price, say, the close price. Using an exponential moving average minimizes the whipsaws that can occur when we compare the close price to the bands. For example, the close price may cross over or under the bands repeatedly without it representing a significant movement, however, using an exponential moving average reduces the possibility of an insignificant cross over or under.

Figure 2-13 shows a chart with this system applied. Notice that the market is in a bullish trend, with trend lines representing the upper and lower bands.

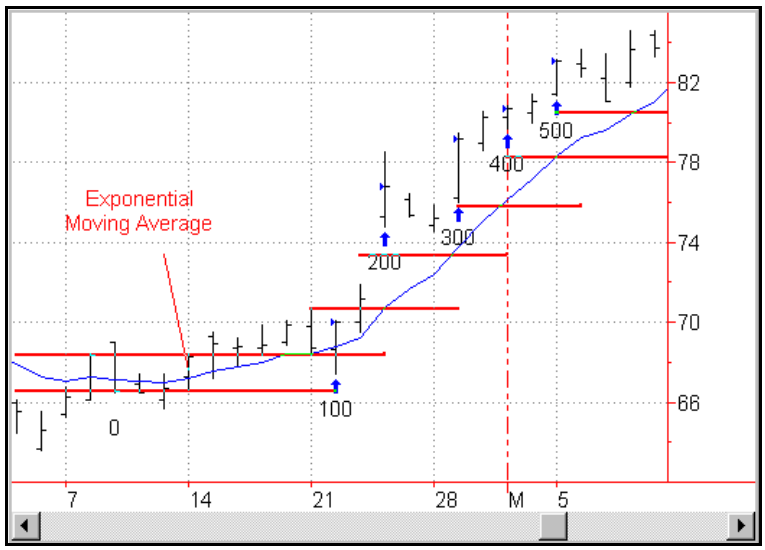


Figure 2-13. Trend Follower applied to a chart.

In Figure 2-14, the system was in a long position, which was then closed out as soon as the price retraced to the bottom band. Then, the exponential average crossed under the bottom band for the first time; however, no signal is generated. This is because the medium moving average was still higher than the slow moving average, thus not meeting our criteria for a bearish market.

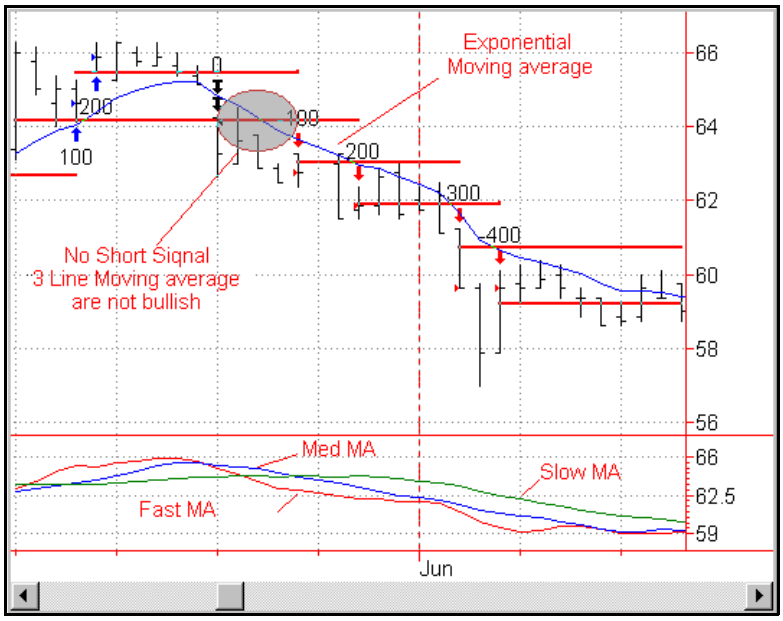


Figure 2-14. The system applied to a chart—notice that although the exponential average crossed, no signal was generated

What differentiates this system is that we want it to pyramid. In other words, if the market we are following is trending, and we are pretty sure about the direction in which it is trending, we will increase the number of shares or contracts in our current position in an attempt to increase our profit.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some set up work, which involved calculating the moving averages and exponential moving averages as well as the top and bottom bands. Our setup, entries and exits are as follows:

Setup

a) Calculate a fast moving average using the last 4 closes, a medium moving average using the last 9 closes, and a slow moving average using the last 18 closes.

b) When the fast moving average is greater than the medium moving average, and the medium moving average is greater than the slow moving average, we consider the market to be in a bull trend.

c) When the fast moving average is less than the medium moving average, and the medium moving average is less than the slow moving average, we consider the market to be in a bear trend.

d) Calculate a 4-bar exponential moving average of the close price.

e) On the first bar, define a top band as the open of the current bar plus an 8-bar volatility times a factor of 2.

f) On the first bar, define a bottom band as the open of the current bar minus an 8-bar volatility times a factor of 2.

g) When the exponential moving average crosses over the top band, the original top band becomes the new bottom band and a new top band is defined by adding the volatility of the last 8 bars times a factor of 2 to the new bottom band.

h) When the exponential moving average crosses under the bottom band, the original bottom band becomes the new top band and a new bottom band is defined by subtracting the volatility of the last 8 bars times a factor of 2 to the new top band.

Long Entries

i) If the market is considered to be in a bull trend, and the exponential moving average crosses over the top band, then buy this bar at the close.

Short Entries

j) If the market is considered to be in a bear trend, and the exponential moving average crosses under the bottom band, then sell this bar at the close.

Exit Orders

k) Exit out of all long trades when the price retraces to any bottom band.

l) Exit out of all short trades when the price breaks any top band.

m) Place a money management protective stop.

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: Trend Follower System

```
Inputs: XMALen(4), VolLen(8), MultFact(2);
Vars: TopBand(0), BotBand(0), XAvg(0), FastMA(0), MedMA(0), SlowMA(0),BullTrend(false),
        BearTrend(false);

{ Define initial Top and Bottom Band }
If BarNumber = 1 then Begin
        TopBand = Open + Volatility( VolLen ) * MultFact;
        BotBand = Open - Volatility( VolLen ) * Multfact;
End;

{ Define market mode with 3-Line Moving Averages }
XAvg = XAverage( Close , XMALen );
FastMA = Average( Close, 4 );
MedMA = Average( Close, 9 );
SlowMA = Average( Close, 18 );
BullTrend = FastMA > MedMA AND MedMA > SlowMA;
BearTrend = FastMA < MedMA AND MedMA < SlowMA;

{ Place long entry orders }
If XAvg > TopBand AND BullTrend then Begin
        Buy this bar on Close;
        BotBand = TopBand;
        TopBand = TopBand + Volatility( VolLen ) * MultFact;
End;

{ Place short entry orders }
If XAvg < BotBand AND BearTrend then Begin
        Sell this bar on Close;
        TopBand = BotBand;
        BotBand = BotBand - Volatility( VolLen ) * MultFact;
End;

{ Place long and short exit orders }
If MarketPosition = 1 then
        ExitLong next bar at BotBand Stop;
If MarketPosition = -1 then
        ExitShort next bar at TopBand Stop;
```

Inputs

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

Input	Default Value	Description
XMALen	4	Number of bars used to calculate the exponential moving average.
VolLen	8	Number of bars used to calculate recent volatility levels.
MultFact	2	Factor by which to multiply the volatility in order to calculate new top and bottom bands.

In addition to the inputs, we declared several variables, as follows:

```
Vars: TopBand(0), BotBand(0), XAvg(0), FastMA(0), MedMA(0), SlowMA(0),BullTrend(false),
BearTrend(false);
```

The following sections discuss how the long and short entries, as well as the exits, are calculated.

Setup

On the very first bar, the system defines both a top and bottom band using the open of the current bar plus the 8-bar volatility (times a multiplier) for the top band and the open of the current bar minus the 8-bar volatility (times a multiplier) for the bottom band:

```
{ Define initial Top and Bottom Band }
If BarNumber = 1 then Begin
        TopBand = Open + Volatility( VolLen ) * MultFact;
        BotBand = Open - Volatility( VolLen ) * MultFact;
End;
```

Long Entries

When the exponential moving average of the closing prices is greater than the top band and the market is in a bull trend, a buy order is placed at the close of the current bar. Also, the original top band becomes the new bottom band and a new top band is calculated:

```
{ Place long entry orders }
If XAvg > TopBand AND BullTrend then Begin
        Buy  this bar at close;
        BotBand = TopBand;
        TopBand = TopBand + Volatility( VolLen ) * MultFact;
End;
```

Short Entries

When the exponential moving average of the closing prices is less than the bottom band and the market is in a bear trend, a sell order is placed at the close of the current bar. Also, the original bottom band becomes the new top band and a new bottom band is calculated:

```
{ Place short entry orders }
If XAvg < BotBand AND BearTrend then Begin
        Sell this bar at close;
        TopBand = BotBand;
        BotBand = BotBand - Volatility( VolLen ) * MultFact;
End;
```

Exit Orders

Once a long position is established, the system places a stop order at most recent bottom band. Likewise, when a short position is established, the system places a sell stop order at the most recent top band:

```
{ Place long and short exit orders }
If MarketPosition = 1 then
        ExitLong next bar at BotBand Stop;
If MarketPosition = -1 then
        ExitShort next bar at TopBand Stop;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for the default number of contracts to trade, commission and slippage. We did not include margin because we designed this system for stocks.

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. Unless specified within the system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) In the **Properties** tab, we selected the option **Allow Multiple Entries in the same direction by same and different entry orders**. This is because this system is designed to scale into a position.

d) Also in the **Properties** tab, we typed the number 4 in the option **Maximum number of entries per position**. This limits the number of times the system can enter the market in the same direction to 4. For example, if we type 4 in this option and the system is buying blocks of 100 shares, the system to hold a maximum of 400 shares at any given time.

Testing & Improving

Figure 2-15 shows a sample System Report for the Trend Follower system:

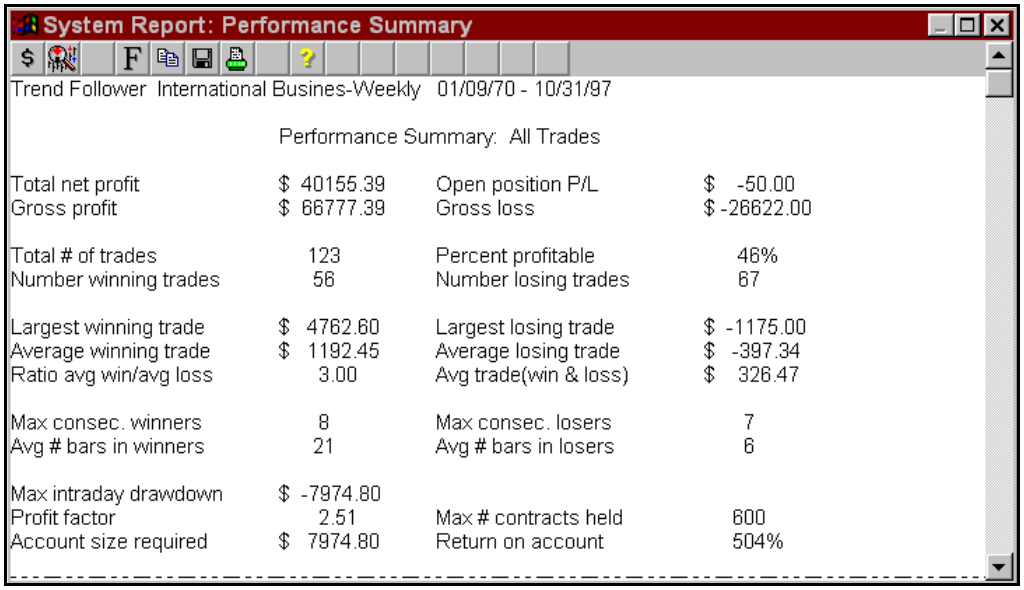


Figure 2-15. System Report for the Trend Follower system

Again, the System Report is typical for a trending system. Notice the low percentage of profitable trades and the high ratio of average winning trades versus average losing trades. Also notice that the average trade (win & loss) is low corresponding to the average winning trade; this is due to the large percentage of losing trades (in this case 54%).

Suggestions for Improvement

One item we can play with is the value used for the input MultFact, which determines the width of the channel. When we reduce the value of this input, the system becomes more sensitive and our positions will increment at a faster pace. However, the system will also be more susceptible to noise in the market. Therefore, we will adjust this input depending on the market we are trading in order to find a satisfactory balance between sensitivity and noise.

This system attempts to ride the movement of the price, either up or down. However, it uses only one filter—three moving averages. Therefore, we might want to use the pyramid technique this system uses in conjunction with another entry and confirmation technique. For example, we may want to use the technique used by the Trend Line Automatic System to enter the market initially and then use the technique used by the Trend Follower to determine when increment the position and exit.

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 up-trend or buying a market that is in a down-trend. 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 that differ in their approach but are all designed to make the most of the sideways movement of a market.

In This Chapter

Spread

Some markets seem to have a strong correlation, that is, the price movement of one market has a strong relationship with that of another market, index, or other data stream. We designed this Spread system in order to follow the percent change of the market we were trading versus another data stream.

We began with the idea that if the market is moving away from the referenced data stream—that is, there is significant divergence—the market is probably going to return to normal levels in the near future. If the spread reaches abnormally high levels, we consider the market to be overbought; and if the spread reaches abnormally low levels, we consider the market to be oversold.

For instance, say we have seen a marked correlation between the Dow Jones Industrial Average and a stock we are trading. When the percent change of the stock for the last 10 trading days is much greater than the percent change of the Dow (meaning that the spread increases), then we would want to enter the market.

Figure 3-1 shows a chart window that contains two markets, the market we're trading and an index (plotted in a hidden subgraph). We applied the Spread system as well as a spread indicator that we created, called Spread System Ind, so we could visually follow the spread line. Notice that when the spread line crosses over the overbought band we specified, the system generates a short signal. Conversely, when the spread line crosses under the oversold band we specified, the system generates a buy order.

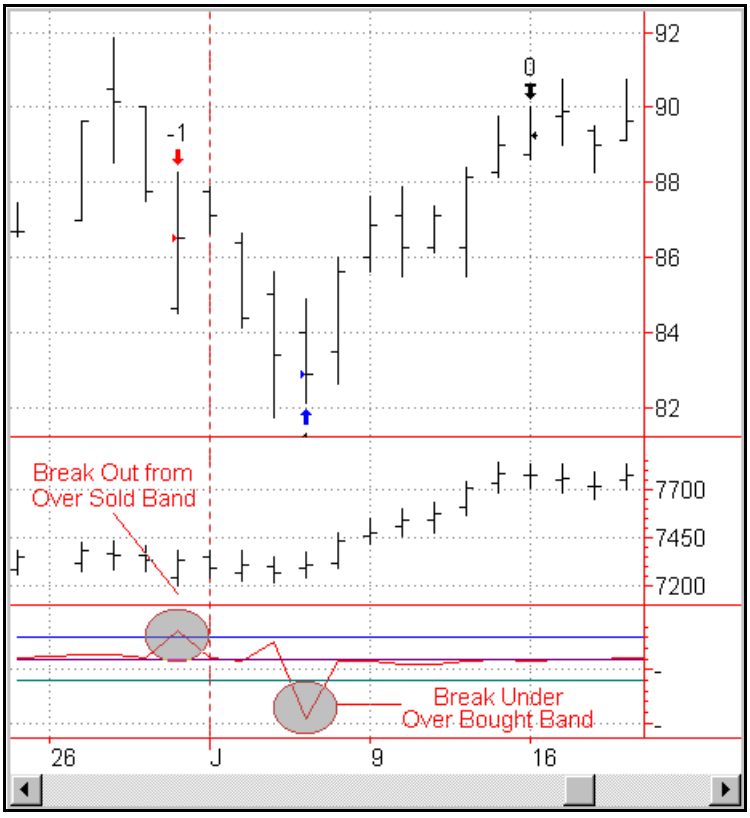


Figure 3-1. Spread system applied to a chart along with Spread indicator

An important component of this system is the level defined for the overbought and oversold bands. We used inputs for the levels, this way, we can change them depending on the market we're trading and how sensitive we want the system to be.

To exit from our positions, we'll use trailing stops, but we'll wait for the position to generate some profit before placing the stops. We are looking to buy at the support and sell at the resistance, and when we place a trailing stop, we want to capture profits; therefore, we will want to wait for the position to generate a profit first.

To place the trailing stops, we will use our standard criteria and exit out of our long positions at the lowest low of the last 6 bars and out of our short positions at the lowest low of the last 6 bars. Again, we will place these orders only once we've registered a profit, and we'll use an input and add or subtract an extra measure to the highest high or from lowest low, respectively.

Plus, since this is a Support & Resistance system, it is possible for the system to be in a stale position. That is, in a position that is not necessarily losing money, but it's not generating any profit either. Therefore, we'll place exit orders once we've been in a position more than six bars and the profit is less than 100 dollars.

As a final method of risk control, we are going to use a money management stop. The level or amount we specify for this stop will depend on how much money we have invested.

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 percent change values and the spread. The setup, entries and exits are described next:

Setup

a) Calculate the percent change of the traded instrument:

```
Data1 Percent Change = ( Close of current bar / Close of 10 bars ago  -  1 ) * 100
```

b) Calculate the percent change of the index or related instrument:

```
Data2 Percent Change = ( Close of current bar / Close of 10 bars ago  -  1 ) * 100
```

c) Calculate the relative strength of the percent change of the traded instrument versus the percent change of the index:

```
Spread = ( Data1 Percent Change / Data2 Percent Change  -  1 ) * 100
```

Long Entries

d) If the spread is less than the oversold level (92), then buy this bar on the close.

Short Entries

e) If the spread is higher than the overbought level (108), then sell this bar on the close.

Exits

f) Exit out of all long positions when the market retraces to the lowest low of the last 6 bars.

g) Exit out of all short positions when the market breaks out of the highest high of the last 6 bars.

h) If 6 bars have gone by without any significant profit, exit all positions.

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

```
Inputs: Length(10), OBLevel(108), OSLevel(92), TrailBar(6), LowProf(100);
Vars:  PctChange1(0), PctChange2(0), Spread(0), ExtraMeasure(0);

{ Percent Change of Data1 and Data2 }
If Close[Length] of Data1 <> 0 then
        PctChange1 = ( Close of Data1 / Close[Length] of Data1 ) * 100;
If Close[Length] of Data2 <> 0 then
        PctChange2 = ( Close of Data2 / Close[Length] of Data2 ) * 100;

{ Strength of Data1 versus Data2 }
If ( PctChange2 ) <> 0 then
        Spread = ( PCtChange1 / PctChange2 ) * 100;

{ Entry orders }
If Spread < OSLevel Then
        Buy this bar at Close;
If Spread > OBLevel Then
        Sell this bar at Close;

{ Trailing stop }
If PositionProfit > 0 then Begin
        If MarketPosition = 1 then
                ExitLong ("LongTrailStop") next bar at Lowest( Low , TrailBar ) - ExtraMeasure stop;
        If MarketPosition = -1 then
                ExitShort ("ShortTrailStop")next bar at Highest( High , TrailBar ) + ExtraMeasure stop;
End;

{ Stale position exit }
If BarsSinceEntry > 6 AND PositionProfit < LowProf then Begin
        ExitLong this bar on Close;
        ExitShort this bar on Close;
End;
```

Inputs

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

Input	Default Value	Description
Length	10	The number of trailing bars used to calculate the percent change for both data streams.
OBLevel	108	The overbought or resistance level at which the system will consider the instrument as overbought and prone to suffer a fall in price.
OSLevel	92	The oversold or support level at which the system will consider the instrument as oversold and prone to have a bullish rally.

Input	Default Value	Description
TrailBar	6	Number of bars to use when finding the highest high price or lowest low price in order to place trailing stop orders.
LowProf	100	Dollar amount considered as low profit before a position is terminated due to lack of movement by the instrument.

In addition to the inputs, we declared several variables, as follows:

```
Variables: j(0), SH(False), SL(False), SwingHAvg(0), SwingLAvg(0), HiReady(False), LoReady(False),
noTLDrawn(true), AddExitPrice(0), LowestLo(0), HighestHi(0);
```

The following sections discuss how the set up, long and short entries as well as the exits are calculated.

Setup

We calculate the percent change of Data1 (the intrument we are trading) and Data2 (the index or instrument to which we are comparing Data1). Note that before the percent changes are calculated, we check to make sure we aren't dividing by zero:

```
If Close[Length] of Data1 <> 0 then
        PctChange1 = ( Close of Data1 / Close[Length] of Data1 ) * 100;
If Close[Length] of Data2 <> 0 then
        PctChange2 = ( Close of Data2 / Close[Length] of Data2 ) * 100;
```

Once the percent changes of Data1 and Data2 are calculated, the system will calculate the spread, or relative strength of Data1 as compared to Data2:

```
If ( PctChange2 ) <> 0 then
        Spread = ( PCtChange1 / PctChange2 ) * 100;
```

Long Entries

When the spread falls below the over sold level, the system will generate a buy signal:

```
If Spread < OSLevel Then
        Buy this bar at Close;
```

Short Entries

When the spread rises above the over bought level, the system will generate a sell signal:

```
If Spread > OBLevel Then
        Sell this bar at Close;
```

Exits

For the long side, we will place a trailing stop at the lowest low of the last Trailbar bars. For the short side, we will place a trailing stop at the highest high of the last Trailbar bars:

```
If PositionProfit > 0 then Begin
    If MarketPosition = 1 then
        ExitLong ("LongTrailStop") next bar at Lowest( Low , TrailBar ) - ExtraMeasure Stop;
    If MarketPosition = -1 then
        ExitShort ("ShortTrailStop")next bar at Highest( High , TrailBar ) + ExtraMeasure Stop;
End;
```

The divergence between the traded instrument and the index should be relevant to the price of the traded instrument. If there has been no significant price movement within a certain amount of time after a divergence occurs, other events might be affecting the price movement of the traded instrument. Therefore, if a position has been open for more than 6 bars and no significant

profit has been made and it hasn't been closed by one of the stop losses, then the system will exit out of the position:

```
If BarsSinceEntry > 6 AND PositionProfit < LowProf then Begin
        ExitLong this bar on Close;
        ExitShort this bar on Close;
End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for the default number of contracts to trade, commission and slippage. We will include an amount for margin when applicable.

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. Unless specified within the system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in 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 3-2 shows a sample System Report for the Spread system applied to a daily chart.

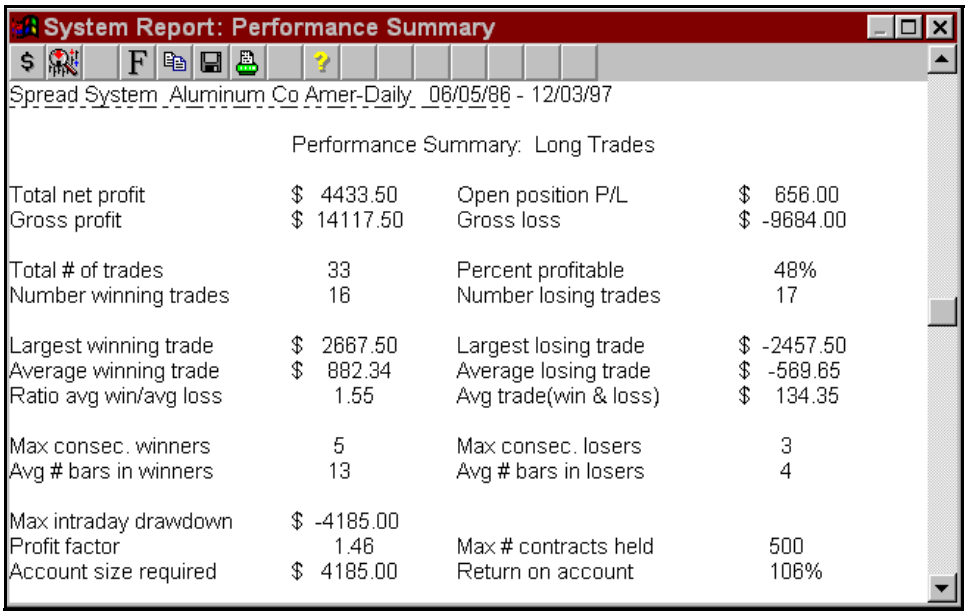


Figure 3-2. Sample System Report for Spread system

The resulting system report is typical of a Support & Resistance system. Notice how the percent of wining trades is slightly over 50 and the average win versus loss ratio is better than 1 yet not as high as the trending systems we looked at in Chapter 2.

Suggestions for Improvement

Support & Resistance systems are easier to trade but will produce many false signals when the market is trending—it will try to short an up-trending market and buy during a down trend. To minimize our losses when the market trends, we could try the following:

a) We could incorporate indicators such as moving averages ADX or DMI, to see if the market is trending, and if it is, don't place trades unless the spread becomes extremely oversold or overbought. In other words, we could move our oversold and overbought bands away from 108 and 92 when the market is trending.

b) We could wait to enter the market when the spread crosses one of the bands and then changes direction. This will enter us into the market one bar too late, but it may filter out some false signals. To do so, we could use the following IF-THEN statement:

```
If Spread[1] > OBLevel AND Spread < Spread[1] Then...
```

c) We could try implementing exits to take us out of possible bad trades. For example, if the spread triggers a buy signal and then reaches 92 and the trade has not produced profits, then we could deem the trade as bad and exit out of it.

Support and Resistance Levels

This system represents a more traditional approach to finding support and resistance levels. This system defines support and resistance levels using significant swing lows and swing highs, respectively. This system finds the last 3 swing high bars and calculates an average of the high prices. This system also finds the last 3 swing lows and calculates an average of the low prices. Horizontal lines are drawn at these support and resistance levels.

A swing high is a bar that is preceded by *x* or more bars with lower or equal highs and followed by *x* or more bars with lower highs.

A swing low is a bar that is preceded by *x* or more bars with higher or equal lows and followed by *x* or more bars with higher lows.

The system then determines the highest high of these 3 swing highs and the lowest low of the last 3 swing lows, and makes sure that the resistance line is in the top third of the distance between the highest high and the lowest low, and that the support line is in the bottom third of this distance. It also makes sure that the distance between the support and resistance lines is at least twice the average true range of the last 10 bars. Figure 3-3 illustrates this concept:

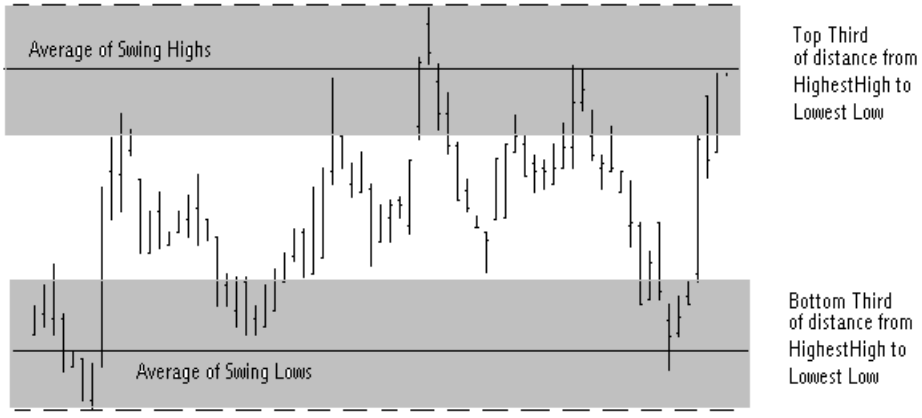


Figure 3-3. Support and Resistance Levels concept

The system then calculates a 4-bar exponential average of the high prices and a 4-bar exponential average of the low prices. These exponential averages are compared to the resistance and support levels, respectively. A sell order is placed when the 4-bar exponential average of the high prices crosses under the average of the swing high prices, and a buy order

is be placed when the 4-bar exponential average of the low prices crosses over the average of the low prices. A 4-bar exponential average is used instead of a price, for example, a close price, in order to eliminate as much 'noise' as possible.

The system checks to make sure the resistance line is in the top third of the distance from the significant swing high to the significant swing low, because if it is not, it could be an indication that the market is trending up, and the up-trending market has no significant resistance. Likewise, when the support line is not in the bottom third, this is probably indicates that the market is trending down, and a down-trending market has no significant support.

To exit, we'll use trailing stop orders and whenthe position has resulted in profit, we'll exit out of all long trades at the lowest low of the last 9 bars, and out of all short positions at the highest high of the last 9 bars. We'll also use a money management stop, which we'll define using the maximum dollar amount loss we can stand to incur per position. Figure 3-4 shows this system applied to a chart.

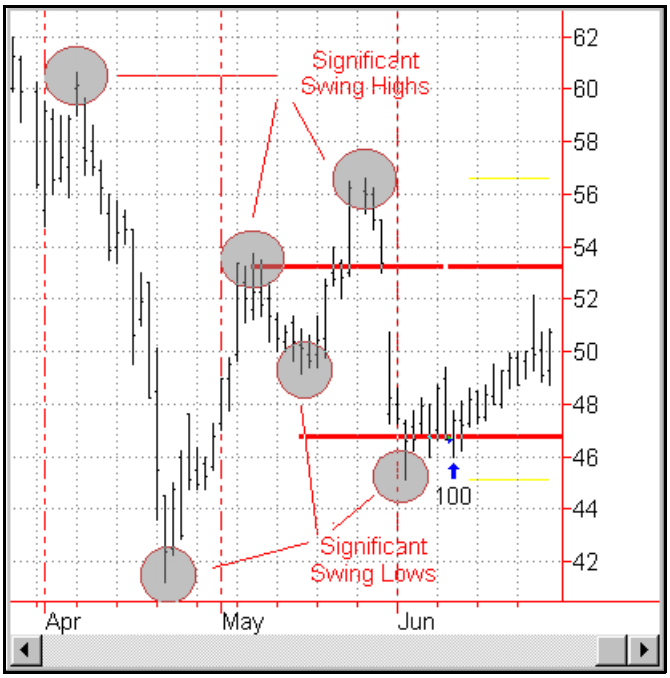


Figure 3-4. Support and Resistance Levels system applied to a chart

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some set up work that involved calculating the swing highs and determining the support and resistance levels. The setup, entries and exits are described next.

Setup

a) Find the last 3 swing highs. A swing high is a bar which has a higher or equal high than the preceding five bars, and a higher high than the following 5 bars.

b) Find the last 3 instances in which a swing low occurred. A swing low is a bar which has a lower or equal low than the preceding five bars, and a lower low than the following 5 bars.

c) Average the prices of the 3 swing highs found. We will call this the resistance level, and draw a horizontal line at this level.

d) Average the prices of the 3 swing lows found. We will call this the support level and draw a horizontal line at this level.

e) Determine if the swing high average is in the top third of the distance between the highest high of the swing highs and the lowest low of the swing lows.

(Swing High Average - Lowest Low) / (Highest High - Lowest Low)

f) Determine if the swing low average is in the bottom third of the distance between the highest high of the swing highs and the lowest low of the swing lows.

(Swing Low Average - Lowest Low) / (Highest High - Lowest Low)

Long Entries

g) If the following conditions are true, place a buy order on the next bar at the open price:

1) The distance from the swing high average to the swing low average is greater than twice the average of the true range of the last 10 bars.

2) The swing high average is in the top third of the distance between the highest high of the swing highs and the lowest low of the swing lows.

3) The swing low average is in the bottom third of the distance between the highest high of the swing highs and the lowest low of the swing lows.

4) The exponential average of the last 4 lows crosses under the resistance level.

Short Entries

h) If the following conditions are true, place a sell order on the next bar at the open price:

1) The distance from the swing high average to the swing low average is greater than twice the average of the true range of the last 10 bars.

2) The swing high average is in the top third of the distance between the highest high of the swing highs and the lowest low of the swing lows.

3) The swing low average is in the bottom third of the distance between the highest high of the swing highs and the lowest low of the swing lows.

4) The exponential average of the last 4 highs crosses over the support level.

Exit Orders

i) Place a money management protective stop order.

j) If an open long position has a profit, exit when the market retraces to the lowest low of the last 8 bars.

k) If an open short position has a profit, exit when the market breaks out of the highest high of the last 8 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: Support and Resistance Levels System

```
Input: HiPrice(High), LoPrice(Low), Strength(5), NumSwing(3), TrailBar(5), LngShrs(100),
ShrtShrs(100);
Variables: j(0), SH(False), SL(False), SwingHAvg(0), SwingLAvg(0), HiReady(False), LoReady(False),
noTLDrawn(true), AddExitPrice(0), LowestLo(0), HighestHi(0);
Arrays: SwingHi[2,20](0), SwingLo[2,20](0);
```

```
{ Finding a Swing High and storing it in the array }
SH = SwingHighBar( 1 , HiPrice , Strength , Strength+1 ) = Strength;
If SH then Begin
        For j = 0 to 19 Begin
                SwingHi[0 , 20 - j] = SwingHi[0 , 19 - j];
                SwingHi[1, 20 - j] = SwingHi[1 , 19 - j];
                SwingHi[2, 20 - j] = SwingHi[2 , 19 - j];
        End;
        SwingHi[0,0] = Date[Strength];
        SwingHi[1,0] = Time[Strength];
        SwingHi[2,0] = HiPrice[Strength];
        If SwingHi[ 1, numswing - 1] <> 0 then
                HiReady = True;
End;

{ FInding a Swing Low and storing it in the array }
SL = SwingLowBar( 1 , LoPrice , Strength , Strength+1 ) = Strength;
If SL then Begin
        For j = 0 to 19 Begin
                SwingLo[0 , 20 - j] = SwingLo[0 , 19 - j];
                SwingLo[1, 20 - j] = SwingLo[1 , 19 - j];
                SwingLo[2, 20 - j] = SwingLo[2 , 19 - j];
        End;
        SwingLo[0,0] = Date[Strength];
        SwingLo[1,0] = Time[Strength];
        SwingLo[2,0] = LoPrice[Strength];
        If SwingLo[ 1, numswing - 1] <> 0 then
                LoReady = True;
End;
LowestLo = 9999; HighestHi = 0;
SwingHAvg = 0; SwingLAvg = 0;
For j = 0 to NumSwing-1 Begin
        SwingHAvg = SwingHAvg + SwingHi[ 2 , j ];
        SwingLAvg = SwingLAvg + SwingLo[ 2 , j ];
        If SwingHi[ 2, j ] > HighestHi then
                HighestHi = SwingHi[ 2 , j ];
        If SwingLo[ 2, j ] < LowestLo then
                LowestLo = SwingLo[ 2 , j ];
End;
SwingHAvg = SwingHAvg / NumSwing;
SwingLAvg = SwingLAvg / NumSwing;
If HiReady and LoReady AND noTLDrawn then Begin
        Value1 = TL_New( SwingHi[ 0 , NumSwing - 1] , SwingHi[ 1 , NumSwing - 1], SwingHAvg,
                SwingHi[ 0 , 0] , SwingHi[ 1 , 0], SwingHAvg);
        Value2 = TL_New( SwingLo[ 0 , NumSwing - 1] , SwingLo[ 1 , NumSwing - 1], SwingLAvg,
                SwingLo[ 0 , 0] , SwingLo[ 1 , 0], SwingLAvg);
        Value99 = TL_SetExtRight( value1 , True );
        Value99 = TL_SetExtRight( value2 , True );
        noTLDrawn = False;
End;
If noTLDrawn = False then Begin
        Value99 = TL_SetEnd(value1, SwingHi[ 0 , NumSwing - 1] , SwingHi[ 1 , NumSwing - 1],
                 SwingHAvg);
        Value99 = TL_SetBegin(value1 ,  SwingHi[ 0 , 0] , SwingHi[ 1 , 0] , SwingHAvg);
        Value99 = TL_SetEnd(value2, SwingLo[ 0 , NumSwing - 1] , SwingLo[ 1 , NumSwing - 1],
                 SwingLAvg);
```

```
        Value99 = TL_SetBegin(value2, SwingLo[ 0 , 0] , Swinglo [ 1 , 0] , SwingLAvg);
End;

If HighestHi - LowestLo <> 0 then Begin
        Value17 = ( SwingHAvg - LowestLo ) /  ( HighestHi - LowestLo );
        Value18 = ( SwingLAvg - LowestLo ) / ( HighestHi - LowestLo );
End;

Condition1 = SwingHAvg - SwingLAvg > Average( TrueRange , 10 )*2;

Condition2 = Value17 > .66 AND Value18 < .33;

Condition3 = Close < SwingHAvg AND Close > SwingLAvg;

If Condition1 AND Condition2 AND Condition3  then Begin
        If XAverage( High , 4 ) Crosses Under SwingHAvg then
                Sell ShrtShrs Shares next bar at Market;
        If XAverage( Low , 4 ) Crosses Over SwingLAvg then
                Buy LngShrs Shares next bar at Market;
End;

If PositionProfit > 0 then Begin
        ExitLong next bar at Lowest( Low , Trailbar ) Stop;
        ExitShort next bar at Highest( High , Trailbar ) Stop;
End;
```

Inputs

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

Input	Default Value	Description
HiPrice	High	Price series used when finding significant swing high points.
LoPrice	Low	Price series used when finding significant swing low points.
Strength	5	Number of bars to the left and right to be used when finding swing high and swing low bars.
NumSwing	3	Number of swing highs and swing lows to use when defining resistance and support levels respectively.
TrailBar	5	Number of bars to consider when placing trailing stop orders at the lowest low and highest high of long and short orders respectively.
LngShrs	100	Number of shares or contracts to buy when a long order is generated.
ShrtShrs	100	Number of shares or contracts to buy when a short signal is generated.

In addition to the inputs, we declared several variables, as follows:

```
Vars:  PctChange1(0), PctChange2(0), Spread(0), ExtraMeasure(0);
```

The following sections discuss how the set up, long and short entries, as well as the exits, are calculated.

Setup

This system will use 2 arrays to track the swing highs and the swing lows. These arrays will hold the information of the last 20 swings, with the most recent swing in row 0 and the oldest swing in row 20:

```
Arrays: SwingHi[2,20](0), SwingLo[2,20](0);
```

As mentioned before, the most recent swing will be stored in row 0 of the array and the oldest in row 20. Every time that a new swing is found, the array will discard the information of the oldest swing, and move the information of each swing one row up. When this is done, the information for the new swing is stored in row 0:

```
{ Finding a Swing High and storing it in the array }
SH = SwingHighBar( 1 , HiPrice , Strength , Strength+1 ) = Strength;
If SH then Begin
        For j = 0 to 19 Begin
                SwingHi[0 , 20 - j] = SwingHi[0 , 19 - j];
                SwingHi[1, 20 - j] = SwingHi[1 , 19 - j];
                SwingHi[2, 20 - j] = SwingHi[2 , 19 - j];
        End;
SwingHi[0,0] = Date[Strength];
SwingHi[1,0] = Time[Strength];
SwingHi[2,0] = HiPrice[Strength];
If SwingHi[ 1, NumSwing - 1] <> 0 then
        HiReady = True;
End;
```

The last instruction before the end of the block IF-THEN statement (the instruction in the bold font above) will set the variable HiReady to true as soon as there are enough swing highs to establish a resistance level. The system performs the same procedures with the swing lows. Next, the system finds the highest swing high, the lowest swing low, the average of the swing highs, and the average of the swing lows:

```
LowestLo = 9999; HighestHi = 0;
SwingHAvg = 0; SwingLAvg = 0;
For j = 0 to NumSwing-1 Begin
        SwingHAvg = SwingHAvg + SwingHi[ 2 , j ];
        SwingLAvg = SwingLAvg + SwingLo[ 2 , j ];
        If SwingHi[ 2, j ] > HighestHi then
                HighestHi = SwingHi[ 2 , j ];
        If SwingLo[ 2, j ] < LowestLo then
                LowestLo = SwingLo[ 2 , j ];
End;
SwingHAvg = SwingHAvg / NumSwing;
SwingLAvg = SwingLAvg / NumSwing;
```

In the following section, the system will draw trend lines and extend these trend lines to the right in the support and resistance levels. Note that it will attempt to draw the trend lines only if the there have been no trend lines drawn before and the variables HiReady and LoReady are true:

```
If HiReady and LoReady AND noTLDrawn then Begin
        Value1 = TL_New( SwingHi[ 0 , NumSwing - 1] , SwingHi[ 1 , NumSwing - 1],
        SwingHAvg, SwingHi[ 0 , 0] , SwingHi[ 1 , 0], SwingHAvg);
        Value2 = TL_New( SwingLo[ 0 , NumSwing - 1] , SwingLo[ 1 , NumSwing - 1],
        SwingLAvg, SwingLo[ 0 , 0] , SwingLo[ 1 , 0], SwingLAvg);
        Value99 = TL_SetExtRight( Value1 , True );
        Value99 = TL_SetExtRight( Value2 , True );
        noTLDrawn = False;
End;
```

Once the trend lines are drawn, the variable noTLDrawn will be false, so on every bar thereafter the system will adjust the beginning and ending points of the trend lines according to the most recent swing points found:

```
If noTLDrawn = False then Begin
        Value99 = TL_SetEnd(value1, SwingHi[ 0 , NumSwing - 1] ,
        SwingHi[ 1 , NumSwing - 1], SwingHAvg);
        Value99 = TL_SetBegin(Value1 ,  SwingHi[ 0 , 0] , SwingHi[ 1 , 0] , SwingHAvg);
        Value99 = TL_SetEnd(Value2, SwingLo[ 0 , NumSwing - 1] , SwingLo[ 1 , NumSwing - 1],
        SwingLAvg);
        Value99 = TL_SetBegin(Value2, SwingLo[ 0 , 0] , Swinglo [ 1 , 0] , SwingLAvg);
End;
```

The system will then identify the position the support in resistance levels in respect to the highest and lowest swings found:

```
If HighestHi - LowestLo <> 0 then Begin
        Value17 = ( SwingHAvg - LowestLo ) /  ( HighestHi - LowestLo );
        Value18 = ( SwingLAvg - LowestLo ) / ( HighestHi - LowestLo );
End;
```

Long Entries

Three conditions will have to be met for the system to place orders: first, the distance from the resistance to the support line has to be at least twice the average true range of the last 10 bars, the support line has to be in the lower third of the range between the highest swing high and the lowest swing low (Value18 < .33) and the resistance line has to be in the upper third of the range between the highest swing high and the lowest swing low (Value17 > .66), and last the close of the bar has to be within the support and resistance levels:

```
Condition1 = SwingHAvg - SwingLAvg > Average( TrueRange , 10 )*2;
Condition2 = Value17 > .66 AND Value18 < .33;
Condition3 = Close < SwingHAvg AND Close > SwingLAvg;
If Condition1 AND Condition2 AND Condition3  then Begin
        If XAverage( High , 4 ) Crosses Under SwingHAvg then
                Sell ShrtShrs Shares next bar at Market;
        If XAverage( Low , 4 ) Crosses Over SwingLAvg then
                Buy LngShrs Shares next bar at Market;
End;
```

Exit Orders

Because we are looking to buy at the support and sell at the resistance, and we want to place atrailing stop to capture profits, we will wait for the system to give profit to place the trailing stops.

A trailing stop will be placed in the long side at the lowest low of the last Trailbar bars and on the short side at the highest high of the last Trailbar bars. To protect for losses at the bar of entry or If an order has not given profit, a money management stop will be placed. Please refer to the general system format section below.

```
If PositionProfit > 0 then Begin
        ExitLong next bar at Lowest( Low , Trailbar ) Stop;
        ExitShort next bar at Highest( High , Trailbar ) Stop;
End;
```

General System Format

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. If applicable, we would include margin. We did not change the **Default Contracts** option, we left it at one.

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. In this system, this is specified by the **Default Contracts** option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) In the **Properties** tab, we selected the **Allow multiple entries in same direction by same and different entry signals** option. If the system is in a long position and market conditions generate another long entry order, the order is generated. This is also the case when we're in a short position and market conditions generate another short entry order. Under **Max open entries per position**, we enter the number 4.

Testing & Improving

Figure 3-5 shows the System Report for the Support and Resistance Levels system.

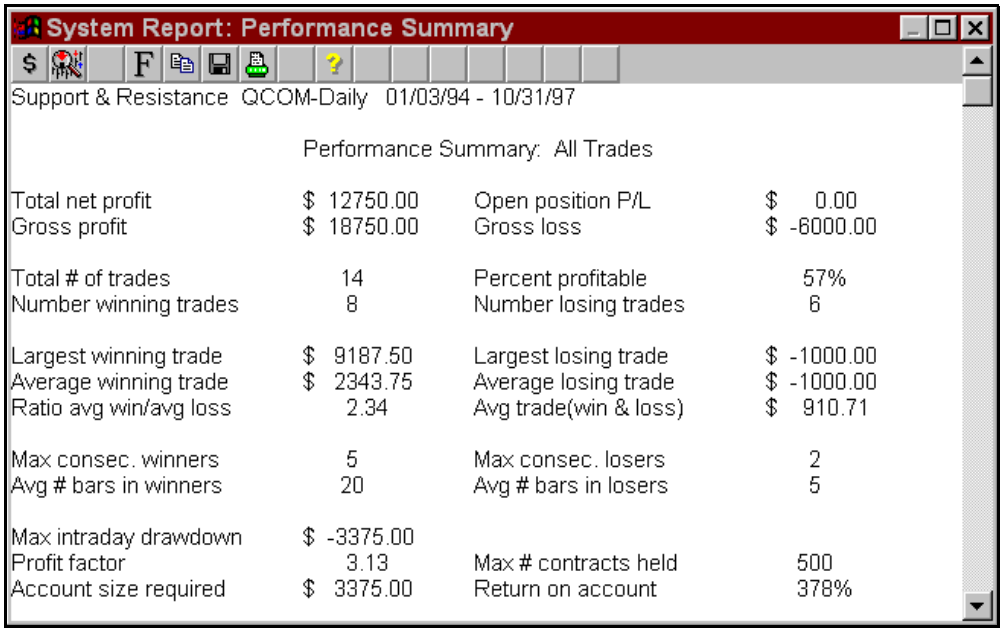


Figure 3-5. Sample System Report for the Support and Resistance Levels system

This System Report is typical for a Support & Resistance system. The percent of profitable trades is over 50% and the ratio of winning versus losing trades is positive (over 1). Something to look out for in the design of this system is the number of trades—the system seems to be very selective in the trades and the report has very few trades.

Suggestions for Improvement

If the system shows strong signs of support and resistance bands, demonstrated by marked swings on both the up and down sides, we may want to increment the input named Strength. When we do so, the system will be more selective when picking significant points for calculating the support and resistance levels.

Also, if the market tends to stay in a price range for longer periods of time, we may want to increment the value of the input named NumSwing. This will make the system consider more significant points when determining the support and resistance levels. Currently, the input defaults to 3, and we may want to consider 4 or 5 points instead.

Aside from changing the inputs, we may want to consider the following ideas:

a) We may want to change the orders to Limit orders. We would buy limit at the resistance level and sell limit at the support level. This will make the system more active but more susceptible to bad signals. To counter this, we could add an extra measure to the support and resistance levels.

b) Instead of just exiting out of a trade when the price action penetrates the support or resistance levels, we could place stop orders to reverse our position if the market starts trending past these levels. So if we are in a short position when the market action reaches the resistance level, but the symbol breaks through the resistance and starts to trend upwards, we will be exited from the short position and entered into a long position.

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 four volatility breakout systems, three of which differ in their approach but are all designed to capture significant change in volatility and limit losses during directionless and/or trending phases.

In This Chapter

Pivot Points

Pivot points is a technique that has been used to trade commodities for many years, and one that many system developers feel it still remains a valid one. There are many ways to calculate pivot points and the support and resistance levels; for the purposes of this system, we are going to use the following method:

Pivot = (Open + High + Low) / 3

Resistance 1 = 2 * Pivot - Low
Resistance 2 = Pivot + Range
Resistance 3 = Resistance 1 + Range

Support 1 = 2 * Pivot - High
Support 2 = Pivot - Range
Support 3 = Support 1 - Range

We determine the pivot by calculating an average of the open, high and low prices of the current bar. Once we obtain the pivot, we can determine the level 1 resistance by multiplying the pivot by 2 and subtracting the low price, the level 2 resistance by adding the range to the pivot, and the level 3 resistance by adding the range to the level 1 resistance. We calculate the support lines in the opposite manner.

We calculate these pivot points using the information from the current bar, and place the orders on the next bar. Traditionally, when you design a volatility breakout system, you place buy stop orders at each one of the resistance levels and sell stop orders at the support levels. However, due to the high volatility of the current market, this would produce too many signals; therefore, we need to make our system more selective as to when it generates orders and at what level.

First, we will disregard support and resistance levels 1. These levels are too close to the market action. Second, we will place orders when the momentum of the market is in our favor. To do this, we will track 3 moving average with different lengths (4-, 9- and 18-day moving averages), and will place long orders only if the faster average is the greater of the three, and the slow average is the lowest. Conversely, we will place short orders when the faster average is the highest. Figure 4-1 illustrates this system on a chart, along with a 3-line moving average indicator.

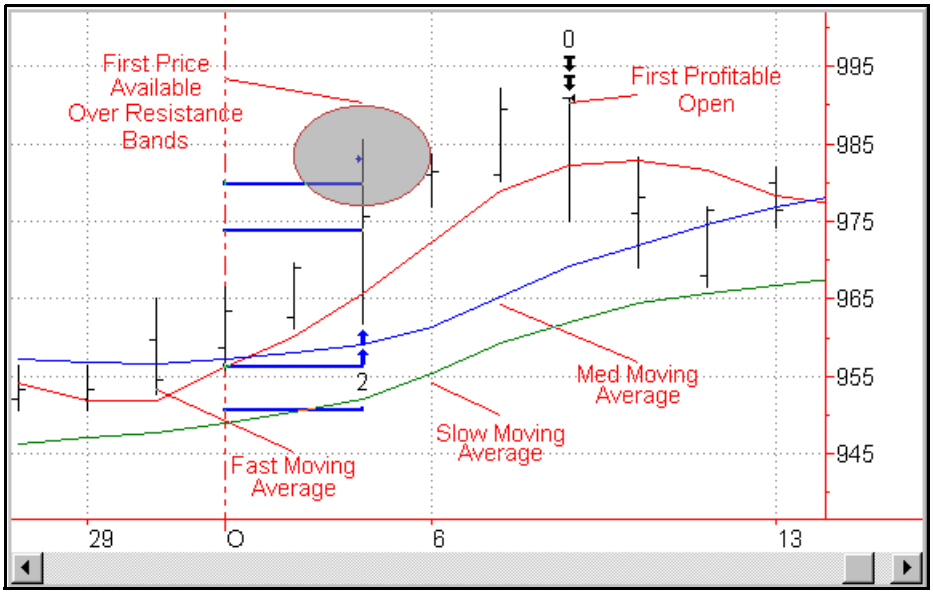


Figure 4-1. Pivot Points system applied to a chart, along with a 3-line moving average indicator

Figure 4-1 shows how a buy order was filled at the first available price above the resistance level.

For our exits, we will exit in a few different ways. First, we will exit all positions, short or long, at the first profitable open. However, in case we do not have a profit, we will place trailing stops that will exit long positions at the lowest low of the last 4 bars and short positions at the highest high of the last 4 bars. Also, we will use a money management stop, the level of which will depend on how much money we are investing.

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 which included calculating the pivot point and the support and resistance lines. How we calculated the setup, entries and exits is described next.

Setup

a) Calculate the pivot points and the support and resistance lines as follows:

```
Pivot = ( Low + High + Close ) / 3;
Resistance1 = ( Pivot * 2 ) - Low;
Resistance2 = Pivot + Range;
Resistance3 = Resistance1 + Range;
Support1 = ( Pivot * 2 ) - High;
Support2 = Pivot - Range;
Support3 = Support1 - Range;
```

b) We will consider the market to be in a bull trend if the fast moving average is greater than the medium moving average, and the medium moving average is greater than the slow moving average.

c) We will consider the market to be in a bear trend if the fast moving average is less than the medium moving average, and the medium moving average is less than the slow moving average.

Long Entries

d) If the market is considered to be in a bull trend, then buy when either the resistance 2 or resistance 3 level is reached.

Short Entries

e) If the market is considered to be in a bear trend, then sell when either support 2 or support 3 level is penetrated.

Exits

f) Exit at the first profitable open of any long or short trade (factoring in commissions).

g) When in a long position, exit if the price retraces to the lowest low of the last 4 bars.

h) When in a short position, exit if the price breaks the highest high of the last 4 bars.

i) Place a money management protective stop.

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: Pivot Points System

```
Input: TrailBar(4);
Vars: Pivot(0), Support1(0), Support2(0), Support3(0), Resistance1(0), Resistance2(0), Resistance3(0),
BearMarket(False) , BullMarket(False);

{ Calculation of Pivot, support and resistance points }
Pivot = ( Low + High + Close ) / 3;
Resistance1 = ( Pivot * 2 ) - Low;
Resistance2 = Pivot + Range;
Resistance3 = Resistance1 + Range;
Support1 = ( Pivot * 2 ) - High;
Support2 = Pivot - Range;
Support3 = Support1 - Range;

{ Determination of market character }
BullMarket = Average( Close , 4 ) > Average( Close , 9 ) AND Average( Close , 9 ) > Average( Close , 18 );
BearMarket = Average( Close , 4 ) < Average( Close , 9 ) AND Average( Close , 9 ) < Average( Close , 18 );

{ Placement of long orders }
If BullMarket then Begin
        Buy ("Bull R2") next bar at Resistance2 Stop;
        Buy ("Bull R3") next bar at Resistance3 Stop;
End;

{ Placement of short orders }
If BearMarket then Begin
        Sell ("Bear S2") next bar at Support2 Stop;
        Sell ("Bear S3") next bar at Support3 Stop;
End;

{ Exit at first profitable open }
If MarketPosition = 1 and Open next bar > EntryPrice + Commission * CurrentContracts/BigPointValue then
        ExitLong ("Long Profit")next bar at Open;
If MarketPosition = -1 and Open next bar < EntryPrice - Commission * CurrentContracts/BigPointValue then
        ExitShort ("Short Profit") next bar at Open;

{ Exit with trailing stops }
ExitLong next bar Lowest( Low , TrailBar ) Stop;
ExitShort next bar Highest( High , TrailBar ) Stop;
```

Inputs

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

Input	Default Value	**Description**
TrailBar	4	Number of bars to consider when placing a trailing stop orders at the lowest low and highest high for long and short orders, respectively.

In addition to the inputs, we declared several variables, as follows:

```
Vars: Pivot(0), Support1(0), Support2(0), Support3(0), Resistance1(0), Resistance2(0),
Resistance3(0), BearMarket(False) , BullMarket(False);
```

The following sections discuss the setup and how the long and short entries, as well as the exits, are calculated.

Setup

The pivot point and the support and resistance levels will be calculated using the following expressions:

```
Pivot = ( Low + High + Close ) / 3;
Resistance1 = ( Pivot * 2 ) - Low;
Resistance2 = Pivot + Range;
Resistance3 = Resistance1 + Range;
Support1 = ( Pivot * 2 ) - High;
Support2 = Pivot - Range;
Support3 = Support1 - Range;
```

The Range function is provided by TradeStation; it takes the low of the bar and subtracts it from the high of the bar.

We will determine the character of the market, bullish or bearish, by making the following comparisons:

```
BullMarket = Average( Close , 4 ) > Average( Close , 9 ) AND Average( Close , 9 ) > Average( Close , 18 );
BearMarket = Average( Close , 4 ) < Average( Close , 9 ) AND Average( Close , 9 ) < Average( Close , 18 );
```

Long Entries

If the market is considered bullish, the system will place orders to enter a long position at either the resistance 2 and resistance 3 level:

```
If BullMarket then Begin
        Buy ("Bull R2") next bar at Resistance2 Stop;
        Buy ("Bull R3") next bar at Resistance3 Stop;
End;
```

Short Entries

If the market is considered bearish, the system will place orders to enter a short position at either the support 2 and support 3 level:

```
If BearMarket then begin
        Sell ("Bear S2") next bar at Support2 Stop;
        Sell ("Bear S3") next bar at Support3 Stop;
End;
```

Exits

This system will exit in a number of different ways. First, we will exit out of any position at the first profitable open. To determine when the first profitable open occurs, we compare the open of the next bar to the entry price, making sure we take into account commissions. Commissions are specified on a per contract/share basis; therefore, we have to multiply the value returned by the Commission function by the number of contracts held (returned by the CurrentContracts function), and then divide this value by the big point value of the instrument. The big point value is the dollar value of a full point move; we can easily obtain it using the BigPointValue function:

Notice that in the same statement we compared the open of the next bar to a value and placed an order for the same open of the next bar.

You cannot reference the open of the next bar unless you're placing orders for the next bar, and you cannot reference the open of the next bar when using multiple data series.

This rule doesn't apply when using the IncludeSystem statement.

```
If MarketPosition=1 and Open next bar > EntryPrice+Commission * CurrentContracts / BigPointValue then
        ExitLong ("Long Profit")next bar at Open;
If MarketPosition=-1 and Open next bar < EntryPrice-Commission*CurrentContracts/BigPointValue then
        ExitShort ("Short Profit") next bar at Open;
```

Second, the system will place a trailing stop at the lowest low of TrailBar bars for long positions and a stop order at the highest high of the last TrailBar bars for short positions.

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

Third, we'll use a money management stop, as discussed in the next section.

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We will also include the margin if applicable.

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. When the system did not specify how many shares/ contracts to handle per transaction, this is determined by the* ***Default Contracts*** *option on this tab.*

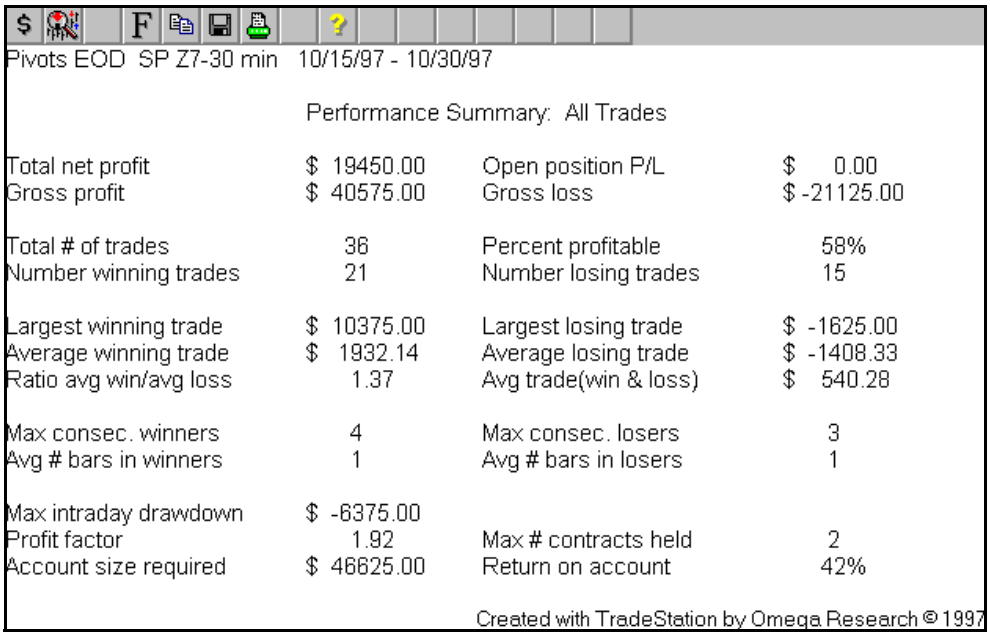
b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

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.*

c) In the **Properties** tab, we selected the **Allow multiple entries in same direction by different entry orders only** option and entered **2** in the **Max open entries per position** edit box. If the system is in a long position and the same conditions that generated the original long entry are met again, the order is ignored; however, if different conditions generate a long entry order, the order is placed. This is also the case when we're in a short position and market conditions generate another short entry order. In this specific system, this will allow the system to enter the market when the price action breaks through resistance level 2 (or support level 2) and then again when level 3 is penetrated.

Testing & Improving

We applied the Pivots Points system to a 30-minute S&P chart; the resulting sample System Report is shown in Figure 4-2.



Pivots EOD SP Z7-30 min 10/15/97 - 10/30/97

Performance Summary: All Trades

Total net profit	$ 19450.00	Open position P/L	$ 0.00
Gross profit	$ 40575.00	Gross loss	$ -21125.00
Total # of trades	36	Percent profitable	58%
Number winning trades	21	Number losing trades	15
Largest winning trade	$ 10375.00	Largest losing trade	$ -1625.00
Average winning trade	$ 1932.14	Average losing trade	$ -1408.33
Ratio avg win/avg loss	1.37	Avg trade(win & loss)	$ 540.28
Max consec. winners	4	Max consec. losers	3
Avg # bars in winners	1	Avg # bars in losers	1
Max intraday drawdown	$ -6375.00		
Profit factor	1.92	Max # contracts held	2
Account size required	$ 46625.00	Return on account	42%

Created with TradeStation by Omega Research © 1997

Figure 4-2. System Report for Pivot Points system

As you can see, it is very accurate as it tries to ride the momentum of the market when it finds unusual activity. For volatility break out systems, it is imperative to factor in commissions and slippage because the system makes many small profits. Notice how the system has 58% profitable trades, yet the ratio of the average win versus loss is only 1.37.

Suggestions for Improvement

If this idea is applied to larger data compression like daily charts, it might become unnecessary (and might hurt the results) to check for the direction of the market before placing orders at the support and resistance levels. So if the entire day or weeks' worth of activity breaks one or both of the pivot points, it will indicate that the momentum of the symbol is going in the direction of the trade.

Pivots Points Intra-Day

The Pivot Points Intra-Day system is essentially the same as the Pivot Points system described previously. It is different only because it is designed specifically to work with intra-day charts. This means we will not calculate the pivot points and the support and resistance levels on every bar, but only on the first bar of the day.

Once we calculate these values for the first bar of the day, we'll use the results throughout the rest of the day. This way, we are letting each day define its own activity range. We used a time compression of at least 10 minutes so that we felt comfortable that enough time had elapsed to define accurately the trading range for the day. We would not want to apply this system to charts plotted with small time compression or bars made up of a small number of ticks; in order to establish the appropriate bands, it requires that the first bar of the day be significant.

We will hold a maximum of two contracts per position. One corresponding to the breakout of the second support or resistance and another corresponding to the breakout of the support or resistance level 3. Figure 4-3 shows this system applied to a chart.

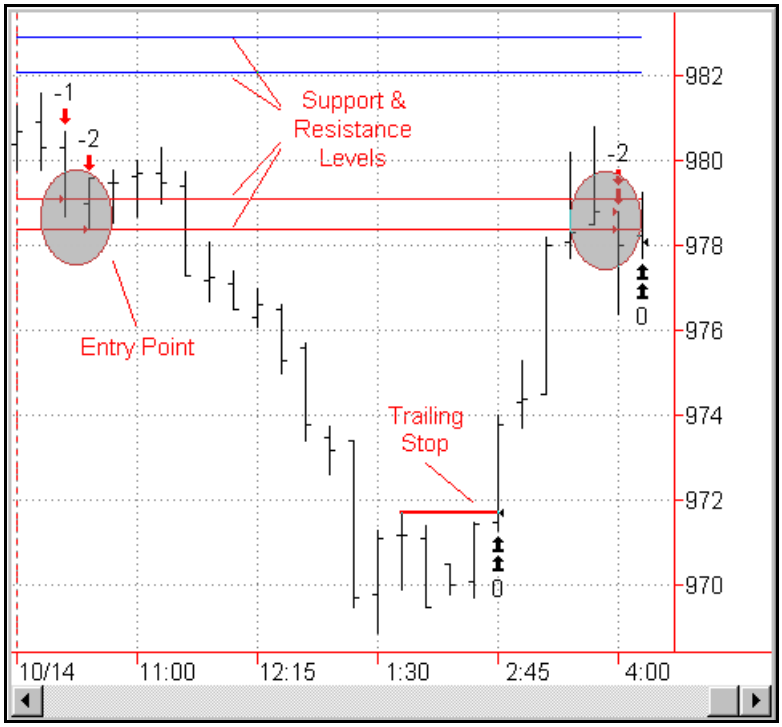


Figure 4-3. The Pivot Points Intra-day system applied to a chart

In terms of exits, we will exit out of any long positions if the price drops to the lowest low of the last 4 bars, and out of any short positions if the price reaches 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. Again, the long and short entries reverse your position, whereas the exits will close out your existing position and exit you from the market. The entries and exits are described next.

Setup

a) Calculate the pivot points and the support and resistance lines using the prices available in the first intra-day period of the trading session:

```
Pivot = ( Low + High + Close ) / 3;
Resistance1 = ( Pivot * 2 ) - Low;
Resistance2 = Pivot + Range;
Resistance3 = Resistance1 + Range;
Support1 = ( Pivot * 2 ) - High;
Support2 = Pivot - Range;
Support3 = Support1 - Range;
```

Long Entries

b) Unless it is the last trading period, buy at the resistance 2 or resistance 3 level or anything higher.

Short Entries

c) Unless it is the last trading period, sell at the support 2 or support 3 level or anything lower.

Exit Orders

d) Exit out of any long positions if the price retraces to the lowest low of the last 4 bars.

e) Exit out of any short positions if the price breaks 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: Pivot Points Intra-day System

```
Inputs: TrailBar(4);
Vars: Pivot(0), Support1(0), Support2(0), Support3(0), Resistance1(0), Resistance2(0), Resistance3(0),
NoLongTradeToday(False), NoShortTradeToday(False);

{ Calculate Pivots at the begining of the day }
If ( Date <> Date[1] ) then Begin
        Pivot = ( Low + High + Close ) / 3;
        Resistance1 = ( Pivot * 2 ) - Low;
        Resistance2 = Pivot + Range;
        Resistance3 = Resistance1 + Range;
        Support1 = ( Pivot * 2 ) - High;
        Support2 = Pivot - Range;
        Support3 = Support1 - Range;
End;

If ( Date next bar = Date ) or ( ( Time > Sess1StartTime ) and ( Time < Sess1EndTime ) ) then Begin
        { Placement of Short orders }
        If Low > Support3 then Begin
                Sell next bar at  Support3 Stop;
                Sell next bar at Support2 Stop;
        End;
        { Placement of long orders }
        If High < Resistance3 then Begin
                Buy next bar at Resistance2 Stop;
                Buy next bar at Resistance3 Stop;
        End;
End;

{ Trailing Stop }
ExitLong next bar at Lowest( Low , TrailBar ) Stop;
ExitShort next bar at Highest( High , Trailbar ) Stop;
```

Inputs

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

Input	Default Value	Description
TrailBar	4	Number of bars to consider when placing a trailing stop orders at the lowest low and highest high of long and short orders respectively.

In addition to the inputs, we declared several variables, as follows:

```
Vars: Pivot(0), Support1(0), Support2(0), Support3(0), Resistance1(0), Resistance2(0), Resistance3(0),
NoLongTradeToday(False), NoShortTradeToday(False);
```

The following sections discuss how the long and short entries, as well as the exits, are calculated.

Setup

The pivot point and the support and resistance levels will be calculated using expressions similar to those used in the previous trading system (Pivot Points), but these values will be calculated only for the first bar of the day. For example, if the system is applied to a 30-minute

chart, the pivots will be calculated using the first 30 minutes of price activity and then the values will be used throughout the day:

```
If ( Date <> Date[1] ) then Begin
        Pivot = ( Low + High + Close ) / 3;
        Resistance1 = ( Pivot * 2 ) - Low;
        Resistance2 = Pivot + Range;
        Resistance3 = Resistance1 + Range;
        Support1 = ( Pivot * 2 ) - High;
        Support2 = Pivot - Range;
        Support3 = Support1 - Range;
End;
```

Long and Short Entry Orders

Once the pivot point and the support and resistance levels are calculated, these values are used throughout the rest of the trading day. The long orders are placed only if the high of the current bar is lower than the resistance 3 level, and the short orders are placed only if the low is greather than the support 3 level. The system looks for breakouts of these bands, and we don't want the system attempting to re-enter the market at there resistance or support levels if the system has already exited with a profit once it has already broken these resistance levels.

```
If (Date next bar = Date) or ( (Time > Sess1StartTime) and (Time < Sess1EndTime) ) then Begin
        If Low > Support3 then Begin
                Sell next bar at  Support3 Stop;
                Sell next bar at Support2 Stop;
        End;
        If High < Resistance3 then Begin
                Buy next bar at Resistance2 Stop;
                Buy next bar at Resistance3 Stop;
        End;
End;
```

Exit Orders

The system will place a trailing stop at the lowest low of TrailBar bars for long positions, and will place a stop order at the highest high of the last TrailBar bars for short positions:

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

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We will also include the margin if applicable.

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. When the system did not specify how many shares/ contracts to handle per transaction, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of the position.

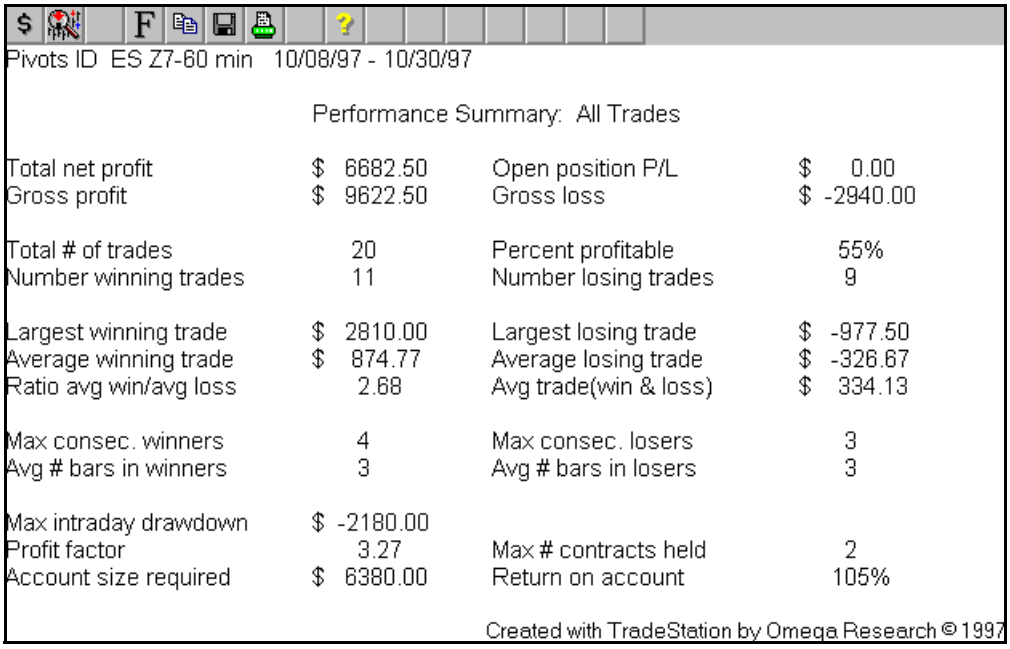
Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are*

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.

c) In the **Properties** tab, we selected the **Allow multiple entries in same direction by different entry orders only** option and entered **2** in the **Max open entries per position** edit box. If the system is in a long position and the same conditions that generated the original long entry are met again, the order is ignored; however, if different conditions generate a long entry order, the order is placed. This is also the case when we're in a short position and market conditions generate another short entry order. In this specific system, this will allow the system to enter the market when the price action breaks through resistance level 2 (or support level 2) and then again when level 3 is penetrated.

Testing & Improving

We applied the Pivot Points Intra-day system to a chart; the sample System Report is shown in Figure 4-4.



Pivots ID ES Z7-60 min 10/08/97 - 10/30/97

Performance Summary: All Trades

Total net profit	$ 6682.50	Open position P/L	$ 0.00
Gross profit	$ 9622.50	Gross loss	$ -2940.00
Total # of trades	20	Percent profitable	55%
Number winning trades	11	Number losing trades	9
Largest winning trade	$ 2810.00	Largest losing trade	$ -977.50
Average winning trade	$ 874.77	Average losing trade	$ -326.67
Ratio avg win/avg loss	2.68	Avg trade(win & loss)	$ 334.13
Max consec. winners	4	Max consec. losers	3
Avg # bars in winners	3	Avg # bars in losers	3
Max intraday drawdown	$ -2180.00		
Profit factor	3.27	Max # contracts held	2
Account size required	$ 6380.00	Return on account	105%

Created with TradeStation by Omega Research © 1997

Figure 4-4. Sample System Report of Pivot Points Intra-day system applied to a 60-minute S&P chart

This report shows the makings of a traditional breakout system, as it has a high percentage of wining trades, yet it has an unusually high average win versus loss ratio. Drawdown is at an acceptable level, and overall the system shows promise but we will test it on additional data as the ratio of average win versus loss seems to be quite high for the idea on which we based the system.

Suggestions for Improvement

This system determines trading ranges for the entire session based on the first bar of the day. This enables the system to find breakouts of these ranges and make profit out of trending days. However, when the market is quiet and has low volatility, the system may not be able to determine a significant trading range using only the first bar of the day.

Therefore, we may want to first verify that the trading range is within "normal" parameters—for example, we may want to take an average of the opening bar's range for the last 3 days and compare this to the range of the first bar of the day—and then adjust the bands if the range of today's first bar is found to differ substantially from the average.

Channel Breakout

This is a volatility expansion system that is designed to take advantage of a market move away from the normal trading range. This system attempts to find instances when the studied market reports unusual activity in price, either up or down, and attempts to get into the market to profit from the sudden move.

This system establishes a normal trading range—determined using the highest high of the last 14 bars and by the lowest low of the last 14 bars—and then places orders when two bars breakout of this trading range in the same direction.

We wait for two bars to breakout instead of just one because many times the price action will move away from its normal trading range, breaking out of the upper or lower band, and then correct itself immediately by returning to this normal trading range. This is referred to as the market finding resistance (when moving up) or support (when moving down). Waiting for the second bar helps confirm that the price is in fact moving away from the channel.

Figure 4-5 shows the system applied to a daily chart and buying at the second cross over of a 14-bar trailing channel.

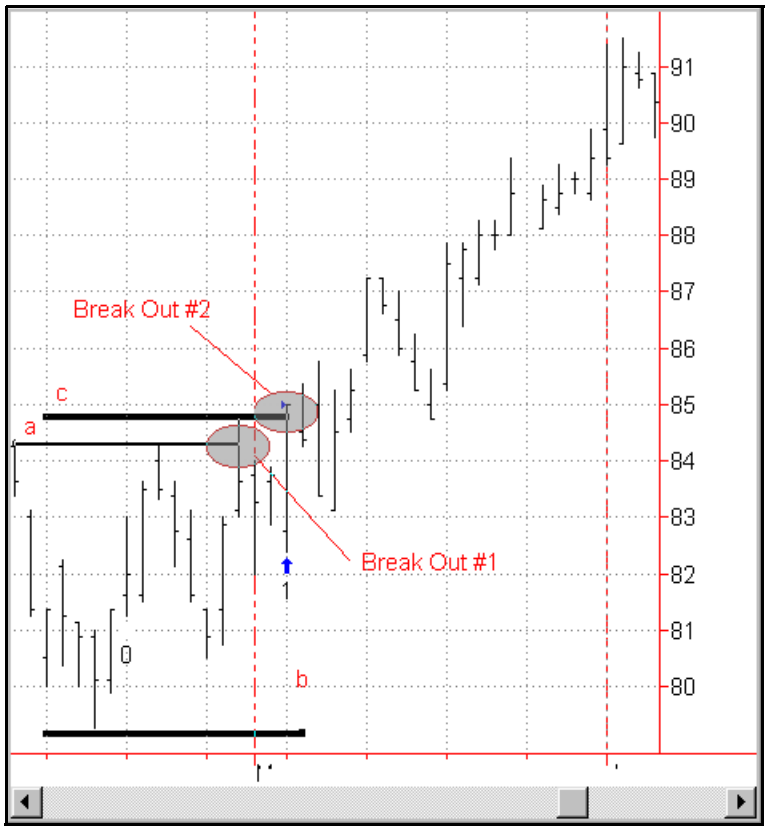


Figure 4-5. The Channel Break Out system applied to a chart

In order to generate a signal, the breakouts have to be consecutive in the same direction but not necessarily on consecutive bars. If one breakout on the up side is found and then the price retraces and breaks on the down side, the up side breakout is discarded. However, if a breakout on the up side is found, the price retraces but doesn't break on the down side and then breaks out on the up side again, a buy signal is generated.

The system will exit out of any long position whenever the lowest low of the last 6 bars is reached, and from short positions whenever the highest high of the last 6 bars is touched by the price.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. There is minimal set up work, which involves calculating the top and bottom values for the channel. The set up, entries and exits are described next.

Setup

a) Calculate the top and bottom values for the trailing channel. For the top value, use the highest high of the last 10 bars. For the bottom value, use the lowest low of the last 10 bars.

Long Entry

b) Buy at the second break out of th 10-bar high trailing channel. These breakouts do not have to be consecutive, but there should be no breakout of the low trailing channel (short entry signal) between the two.

Short Entry

c) Sell at the second break under of the 10-bar low trailing channel. These breakouts do not have to be consecutive, but there should no breakout of a high trailing channel (long entry signal) between the two.

Exit Orders

d) Exit out of any long positions if the market retraces to the lowest low of the last 6 bars.

e) Exit out of any short positions if the market breaks the highest high of the last 6 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: Channel Breakout System

```
Input: Length(10), BrkOuts(2), TrailStp(6);

Vars: HighChannel(0), LowChannel(0), BreakOutCounter(0), BreakUnderCounter(0);

HighChannel = Highest( High , Length )[1];
LowChannel = Lowest( Low , Length )[1];

If High > HighChannel then Begin
        BreakOutCounter = BreakOutCounter+ 1;
        BreakUnderCounter = 0;
End;

If Low < LowChannel then Begin
        BreakUnderCounter = BreakUnderCounter + 1;
        BreakOutCounter= 0;
End;

If BreakOutCounter >= BrkOuts then Begin
        Buy this bar on Close;
        BreakOutCounter = 1;
End;

If BreakUnderCounter >= BrkOuts then Begin
        Sell this bar on Close;
        BreakUnderCounter = 1;
End;

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

Inputs

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

Input	Default Value	Description
Length	10	Number of bars to use to build the channel around the price activity.
BrkOuts	2	Number of breakouts required for the system to places an order.
TrailStp	6	Number of bars to consider when placing a trailing stop orders at the lowest low and highest high of long and short orders respectively.

In addition to the inputs, we declared several variables, as follows:

```
Vars: HighChannel(0), LowChannel(0), BreakOutCounter(0), BreakUnderCounter(0);
```

The following sections discuss how the long and short entries, as well as the exits, are calculated.

Setup

The system stores the value of the high and low channel in the variables HighChannel and LowChannel, respectively. Note that when we use the functions Highest( ) and Lowest( ), we make sure to refer to one bar ago. This is because these functions include the current bar in their calculations and the high of the current bar will never be higher than itself:

```
HighChannel = Highest( High , Length )[1];
LowChannel = Lowest( Low , Length )[1];
```

When a breakout of the high channel is found, the system increments the variable BreakOutCounter by one. This variable serves as a counter and is used to store the number of breakouts the system encounters on the long side. When a breakout of the high channel is found, the system also sets to zero the variable we use as a counter for the breakouts of the low channel (BreakUnderCounter):

```
If High > HighChannel then Begin
        BreakOutCounter = BreakOutCounter + 1;
        BreakUnderCounter = 0;
End;
```

Likewise, when a break under of the low channel is found, the system will increment the counter BreakUnderCounter by one. This variable is used to store the number of break unders that the system encounters on the short side. When a break under of the low channel is found, the system also sets to zero the counter for the break out (BreakOutCounter):

```
If Low < LowChannel then Begin
        BreakUnderCounter = BreakUnderCounter + 1;
        BreakOutCounter = 0;
End;
```

Long Entry

When the variable BreakOutCounter is equal to or greater than the input BrkOuts, it places a long order at the close of the current bar, and resets the long counter to 1:

```
If BreakOutCounter >= BrkOuts then Begin
        Buy this bar on Close;
        BreakOutCounter = 1;
End;
```

Short Entry

Similar to the long side, when the variable BreakUnderCounter is equal to or greater than the input BrkOuts, it places a short order at the close of the current bar, and resets the short counter to 1:

```
If BreakUnderCounter >= BrkOuts then Begin
        Sell this bar on Close;
        BreakUnderCounter = 1;
End;
```

Exit Orders

The system will place a trailing stop at the lowest low of TrailStp bars for long positions, and at the highest high of the last TrailStp bars for short positions:

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

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We will also include the margin if applicable.

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. When the system did not specify how many shares/ contracts to handle per transaction, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) 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.

We have chosen to develop this system without pyramiding, but it will work with pyramiding (or scaling into positions). Let's look at the entry orders on the long side:

```
If High > HighChannel then Begin
        BreakOutCounter = BreakOutCounter + 1;
        BreakUnderCounter = 0;
End;
If BreakOutCounter >= BrkOuts then Begin
        Buy this bar on Close;
        BreakOutCounter = 1;
End;
```

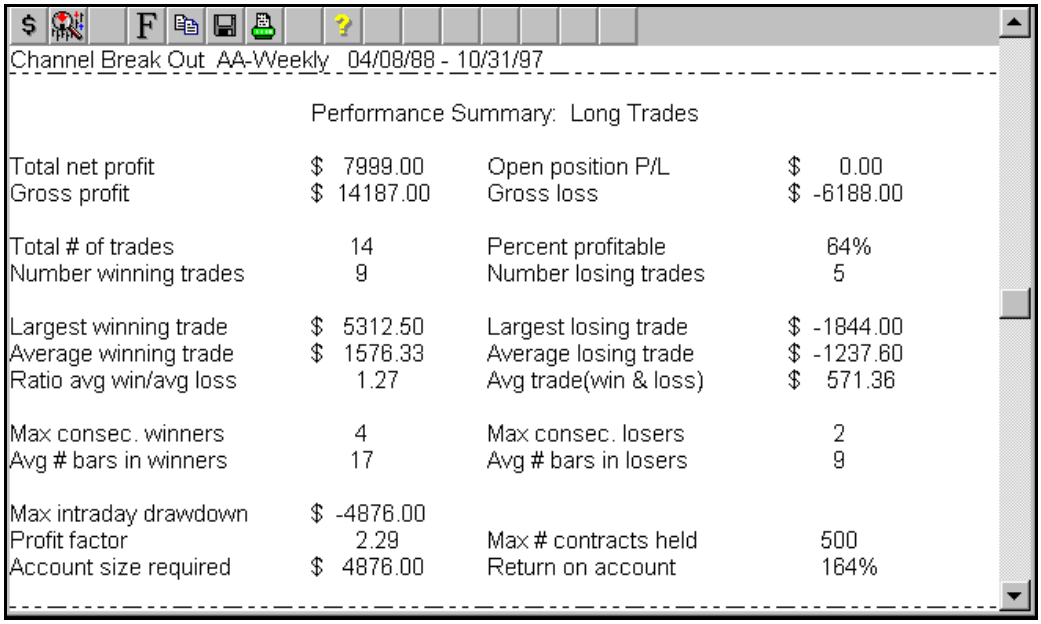
If a breakout is found, the variable BreakOutCounter is incremented, and if a certain number of breakouts have occurred, as specified by the input BrkOuts, a long position will be established. At this point, the breakout counter is set to 1 and will be incremented again if another breakout

is found. The only reason the system doesn't enter a second long position is because we have turned off the pyramiding setting under the **Properties** tab when formatting the system.

However, if we wanted the system to pyramid (scale into a position), we would choose the option **Allow multiple entries in same direction by same and different entry orders** and set the limit for how many times the system will buy (or sell) consecutively by typing a number in the **Maximum number of entries per position** edit box. For example, if the system buys in blocks of 100 shares and we want to limit the system to hold a maximum of 400 shares at a given time, we would type **4** in the edit box.

Testing & Improving

We applied the Channel Break Out system to a chart; the sample System Report is shown in Figure 4-6.



Channel Break Out AA-Weekly 04/08/88 - 10/31/97

Performance Summary: Long Trades

Total net profit	$ 7999.00	Open position P/L	$ 0.00
Gross profit	$ 14187.00	Gross loss	$ -6188.00
Total # of trades	14	Percent profitable	64%
Number winning trades	9	Number losing trades	5
Largest winning trade	$ 5312.50	Largest losing trade	$ -1844.00
Average winning trade	$ 1576.33	Average losing trade	$ -1237.60
Ratio avg win/avg loss	1.27	Avg trade(win & loss)	$ 571.36
Max consec. winners	4	Max consec. losers	2
Avg # bars in winners	17	Avg # bars in losers	9
Max intraday drawdown	$ -4876.00		
Profit factor	2.29	Max # contracts held	500
Account size required	$ 4876.00	Return on account	164%

Figure 4-6. Sample System Report for the Channel Break Out system applied to a chart

This system, like most volatility expansion systems, tries to capture an abrupt market move in a given direction and tries to ride the momentum towards profit. This report shows a very good percentage of profitable trades yet the average wining versus losing trade is much smaller than trending systems—volatility expansion systems try to make profit out of more but less profitable trades.

Because this system uses a trailing stop to exit from a position and take profits, it will tend to stay in the market longer than the usual break out system, which often is looking for any profit to exit and move into the next trade.

Suggestions for Improvement

Markets that are very choppy can break out or under the channel and produce false signals, as shown in Figure 4-7.

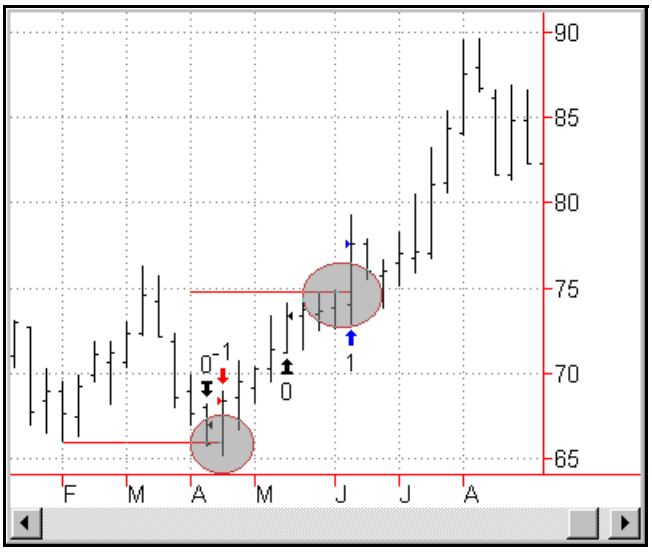


Figure 4-7. Channel Break Out system applied to a choppy market, showing false signals

To have the system become less sensitive to this noise, we can increase the input called BrkOuts. This is the number of breakouts that must occur before the system will generate a long or short signal. The chart in Figure 4-7 has the BrkOuts set to 2 whereas the chart in Figure 4-8 has the input set to 3. Note how it avoided the short entry in the middle of April.

Figure 4-8. Channel Break Out system applied to chart. Input BrkOuts is set to 3 instead of 2

The lower the input, the more sensitive the system will be towards price movement, but the more prone it will be to giving false signals. We will try adjusting this input depending on the market we are testing.

Volatility Expansion (VolEx)

The VolEx system was first developed by Bill Cruz and Charlie Wright in the early ‘90s. It attempts to find days with unusual price activity. When it does so, it enters the market and use the momentum produced by this sudden move to create a quick profit. Like the traditional volatility expansion systems, it will stay in the market very short periods of time, only the time necessary to produce a profit. This system was designed to trade the S&P contract using daily bars, but using its basic principles you can develop a system to be used on any market.

As with any volatility expansion system, the leverage and volatility of the market is important. This type of system normally remains in the market for a few number of bars as it attempts to make its profit out of numerous but relatively less profitable wining trades. In other words, it is developed specifically for use in markets with higher leverage and volatility.

The system will establish a buy and a sell price for every day (or bar). For the buy price, it multiplies the range of the previous day times a factor and adds it to the open price. For the sell price, it multiplies the range of the previous day times a factor but subtracts it from the open price.

Remember, this system is trying to find a significant move in either direction in an attempt to ride the momentum towards profit. Since markets usually fall quicker than they go up, we would need a bigger move down to consider it significant. Therefore, the short factor is normally larger than the buy factor.

This system incorporates several exits. First, will exit either our long or short position on the first profitable open. To do this we will consider commissions, slippage, and a minimum profit level required for the trade. Second, we will exit out of all long positions if the market reaches the lowest low of the last 4 bars, and exit all short positions when the market touches the highest high of the last 4 bars.

The third exit technique is to exit the long or short position if neither of the two previous orders have been met and the system has been in the market for more than 4 bars (days) without producing a minimum profit. In this case, we would want to exit out of the trade, regardless of profit or loss, in order to look for other investment opportunities. Again, the idea of the system is to give you many small profits, there is no reason to remain in the same position without quick profit. Finally, you would place a money management stop to limit the losses per contract. Figure 4-9 shows the VolEx system applied to a chart.

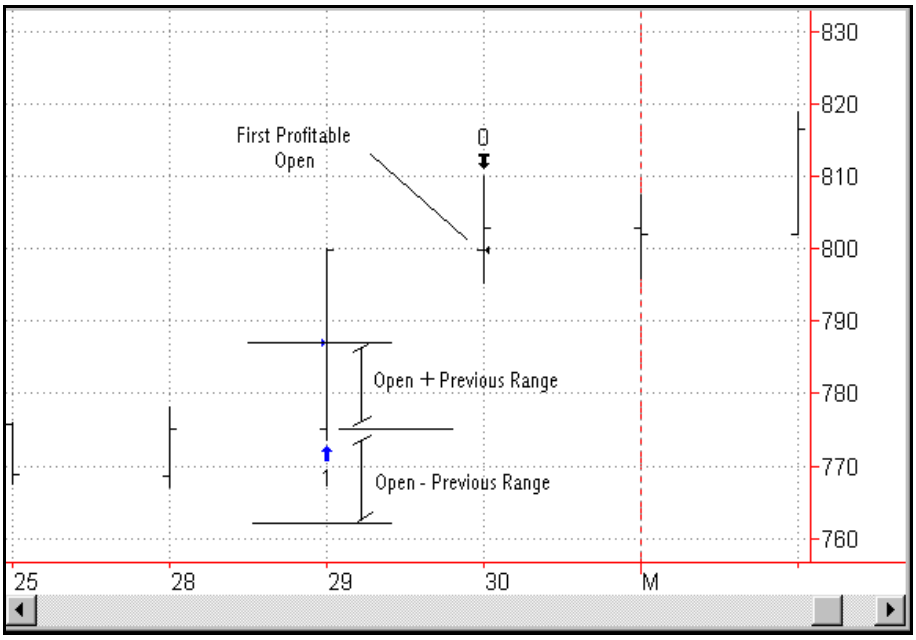


Figure 4-9. VolEx System applied to a chart

Defining your Trading Rules

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

Long Entries

a) Buy on the next day at the open of that bar plus 1.2 times the 4-bar average of the trading range.

Short Entries

b) Sell on the next day at the open of that bar minus 1.8 times the 4-bar average of the trading range.

Exit Orders

c) Exit at the first profitable open of any long or short trade (including commissions).

d) Once in a long position, exit if the price retraces to the lowest low of the last 4 bars.

e) Once in a short position, exit if the price breaks the highest high of the last 4 bars.

f) If 4 bars have elapsed without any of the exits triggered, then exit at the first price of the next trading day.

g) Place a money management protective stop

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: Volatility Expansion System

```
Input: LMult(1.2), SMult(1.8), ProfitPt(0), TrailBar(4);
Vars: BuyLevel(0), SellLevel(0);

{ Entry Orders }
Buy at Open next bar + Average(Range,4) * LMult Stop;
Sell at Open next bar - Average(Range,4) * SMult Stop;

{ Exit at first Profitable Open }
If MarketPosition=1 and Open next bar>EntryPrice+ProfitPt+Commission*CurrentContracts/BigPointValue then
        ExitLong ("Long Profit")next bar at Open;
If MarketPosition=-1 and Open next bar<EntryPrice-ProfitPt-Commission*CurrentContracts/BigPointValue then
        ExitShort ("Short Profit") next bar at Open;

{ Trailing Stops }
If MarketPosition = 1 then
        ExitLong next bar at Lowest( Low , TrailBar ) Stop;
If MarketPosition =-1 then
        ExitShort next bar at Highest( High , TrailBar ) Stop;

{ Stale Position Exit }
If BarsSinceEntry > 4 AND PositionProfit < 500 then Begin
        ExitLong ("Long Too Long") next bar at Market;
        ExitShort ("Short Too Longt") next bar at Market;
End;
```

Inputs

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

Input	Default Value	Description
LMult	1.2	Factor used when adding the average range to the open of the current day to determine the long position entry price.
SMult	1.8	Factor used when subtracting the average range to the open of the current day to determine the short position entry price.
ProfitPt	0	Points that will be used to calculate that a position is profitable when looking for the first profitable open to exit out of any position.
TrailBar	4	Number of bars to consider when placing a trailing stop orders at the lowest low and highest high for long and short orders, respectively.

In addition to the inputs, we declared several variables, as follows:

```
Vars: BuyLevel(0), SellLevel(0);
```

The following sections discuss how the long and short entries, as well as the exits, are calculated.

Long Entries

This system will place an order to initiate a long position if the price breaks out of the open plus the 1.2 times the average range of the last 4 bars of one bar ago. Because TradeStation places stop and limit orders for the next bar, the order would be written as follows:

```
Buy at Open next bar + Average(Range,4) * LMult Stop;
```

When this order is applied to a chart, the order will become active at the moment when the first price of the next trading period becomes available (at the open of the next day if a daily chart is used).

Short Entries

This system will place an order to initiate a short position if the price breaks under of the open minus 1.8 times the average range of the last 4 bars of one bar ago. Because TradeStation places stop and limit orders for the next bar, the order would be written as follows:

```
Sell at Open next bar - Average(Range,4) * SMult Stop;
```

When this order is applied to a chart, the order will become active at the moment when the first price of the next trading period becomes available (at the open of the next day if a daily chart is used).

Exit Orders

This system will exit in a number of different ways. First, we will exit out of any position at the first profitable open. For long positions, we check to see if the open of the next bar is greater than a certain amount, at which point we'll exit at that open. Likewise, for short positions, we check to see if the open of the next bar is less than a certain amount, at which point we'll exit at that open. When calculating the amount to compare to the open, we include the input ProfitPt (minimum required profit in points) and commissions.

Because commissions are specified in a per contract basis, the value returned by the Commission function is multiplied by the number of contracts (obtained using the CurrentContracts function),

and divided by the big point value, which is the dollar value of one whole number movement in price of the instrument (obtained using the function BigPointValue):

```
If MarketPosition=1 and Open next bar>EntryPrice+ProfitPt+Commission*CurrentContracts/BigPointValue then
        ExitLong ("Long Profit")next bar at Open;
If MarketPosition=-1 and Open next bar<EntryPrice-ProfitPt-Commission*CurrentContracts/BigPointValue then
        ExitShort ("Short Profit") next bar at Open;
```

Once a position has been established, the system will look to exit out with a trailing stop. If a long position is open, the system will exit at the lowest low of TrailBar bars. If a short position is open, then the system will exit out of this position at the highest high of the last Trailbar bars.

```
If MarketPosition = 1 then
        ExitLong next bar at Lowest( Low , TrailBar ) Stop;
If MarketPosition = -1 then
        ExitShort next bar at Highest( High , TrailBar ) Stop;
```

The objective of the system is to make profit out of sudden moves. The theory is that eventually this sudden move will wear off and other factors will influence the immediate movement of the market. So if a position is open for more than 4 bars and no significant profit has been made nor has the position been closed by one of the stop losses, then the system will exit out of the position.

```
        If BarsSinceEntry > 4 AND PositionProfit < 500 then Begin
                ExitLong ("Long Too Long") next bar at Market;
                ExitShort ("Short Too Long") next bar at Market;
        End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We will also include the margin if applicable.

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. When the system did not specify how many shares/ contracts to handle per transaction, this is determined by the* ***Default Contracts*** *option on this tab.*

b) 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 dollar amount per contract/share you want to risk before exiting out of 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.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in 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

We applied the Volatility Expansion (Volex) system to a chart with one year of daily S&P data; a sample System Report is shown in Figure 4-10.

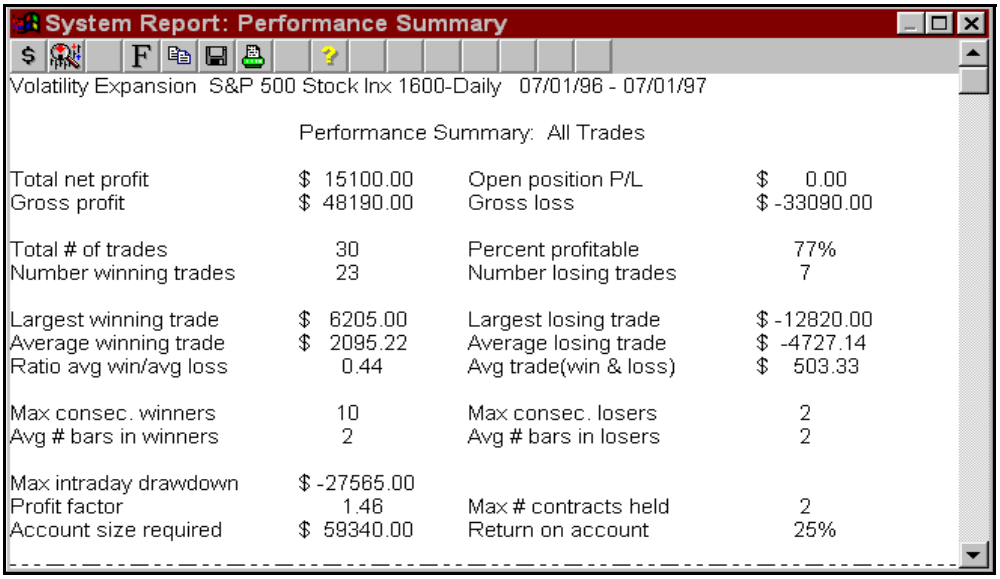


Figure 4-10. Sample System Report for the Volatility Expansion System applied to a chart

This report factors in commission and slippage. The system is remarkably accurate, with close to 77% of winning trades, which compensates for the low average winning versus losing trades ratio of .44. Even though the total net profit is not great at $15,100, consider that it was in the market an average of 2 days for 30 trades throughout the year; this adds up to 60 days in the market and leaves the money available during the rest of the year for other investments.

One problem the system has is the high average of losing trades, which causes a high drawdown. The average losing trade is over $4,000 and the drawdown at $27,000 is almost double the net profit. So, despite the good results, this system might be untradable! It is therefore imperative that we limit losses. Figure 4-11 shows the System Report once we add a money management stop at $1,800 per position.

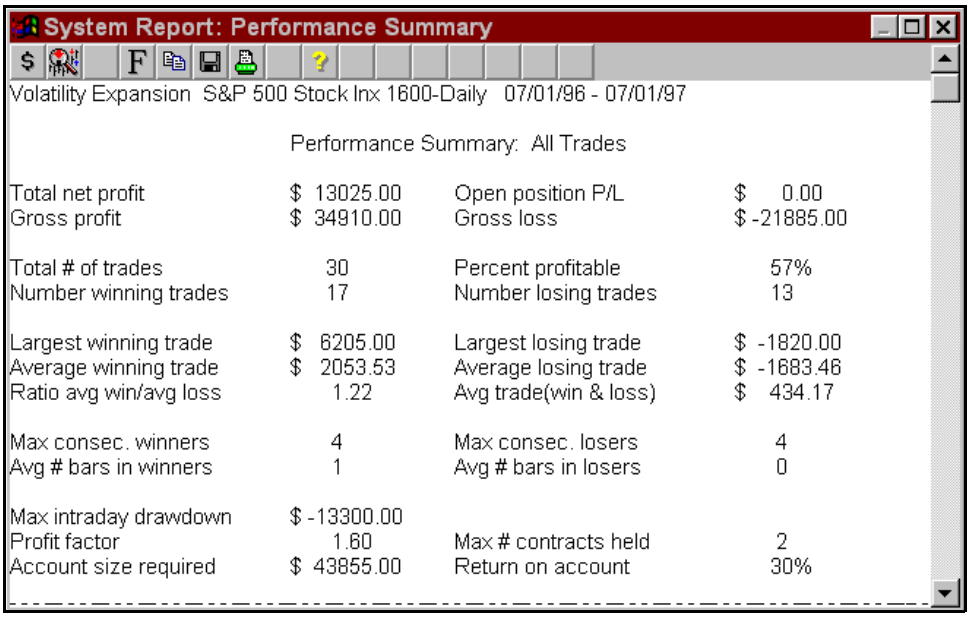
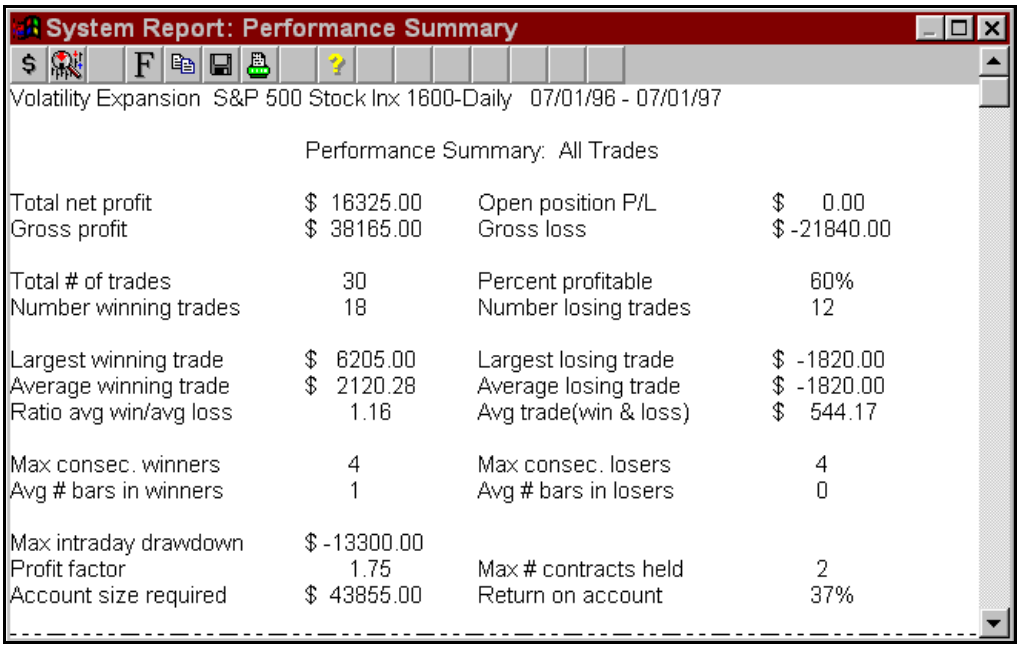


Figure 4-11. Sample System Report for VolEx system after a money management stop is implemented

Adding a money management stop affects net profit in a minor way yet has a relatively strong impact on the percentage of average winning versus losing trades. But the most important changes are the drawdown and the average losing trades—they have dropped to more than half each, making this a system to consider.

Another item we can change is the input ProfitPt. This input represents the additional points of profit to look for in order to exit our trades at the first profitable open. By default, this value is set to 0. Figure 4-12 shows the System Report once we change this number to .25 (a quarter of a point).



System Report: Performance Summary

Volatility Expansion S&P 500 Stock Inx 1600-Daily 07/01/96 - 07/01/97

Performance Summary: All Trades

Total net profit	$ 16325.00	Open position P/L	$ 0.00
Gross profit	$ 38165.00	Gross loss	$ -21840.00
Total # of trades	30	Percent profitable	60%
Number winning trades	18	Number losing trades	12
Largest winning trade	$ 6205.00	Largest losing trade	$ -1820.00
Average winning trade	$ 2120.28	Average losing trade	$ -1820.00
Ratio avg win/avg loss	1.16	Avg trade(win & loss)	$ 544.17
Max consec. winners	4	Max consec. losers	4
Avg # bars in winners	1	Avg # bars in losers	0
Max intraday drawdown	$ -13300.00		
Profit factor	1.75	Max # contracts held	2
Account size required	$ 43855.00	Return on account	37%

Figure 4-12. Sample System Report for VolEx system applied to chart with higher profit requirement

Without affecting drawdown, the net profit and percent of profitable trades have improved. The system is looking for an abrupt move in either direction, so putting the exit further away made the system wait longer to exit. This improved our wining trades a great deal but left our losing trades unchanged.

INDEX

T *(Cont'd)*

V

W