THE DYNAMIC BREAK OUT II STRATEGY

George Pruitt for *Futures Magazine* designed the original Dynamic Break Out system in 1996. This version has done well since it was released for public consumption in 1996. This version will be included in Appendix B. The newer version of the Dynamic Break Out is just like the original, except we have incorporated an additional adaptive filter.

The key to the Dynamic Break Out II system is its ability to adapt its parameters to current market conditions. This system is based on the tried-and-tested Donchian channel system. Remember how the Donchian system works; buy when the high of the day penetrates the highest high price of *x* bars back, and sell when the low of the day penetrates the lowest low of *x* bars back. If you optimize the number of bars to determine your best entry and exit levels, you will discover that different markets work better with different parameters. You will also discover that a particular market goes through different cycles and works better with different parameters through time. For example, the Japanese Yen may have performed better with a look back of 40 days in the 1980s, but now works better with a look back of 20 days. That is the major problem with using a static parameter for all markets. The Dynamic Break Out II system allows the number of look back days to change with the current market. Instead of using a static parameter, this system changes the parameters based on an aspect of the current market.

Before you can use an adaptive parameter, you must come up with a function or adaptive engine that automatically changes the value of the once static parameter. The input of this adaptive engine should be some form of market statistic. In the case of the Dynamic Break Out II, we used market volatility. When market volatility expands, so does the number of look back days in our break out calculation. Increased market volatility usually equates to market indecisiveness. By increasing the number of look back days when market volatility increases, we make it more difficult for the system to initiate a trade. When market volatility decreases, we reduce the number of look back days. Low market volatility equates to a trending market. By decreasing the number of look back days, we encourage the system to initiate a trade. This helps the Dynamic Break Out II to lock into long-term profits and be on the look out for a change in the long-term trend. We used market volatility to fuel our adaptive engine, but you could use any market characteristic. We can visualize an engine that uses a market's overbought/oversold state. If we had a long position in a market, and it became overbought, we could use an overbought/oversold indicator to adapt the parameter that determines the sell point.

Once an adaptive engine is dreamed up and it is pumping out values, you must maintain the values in an acceptable range. The Dynamic Break Out II system will not let the look back days go above 60 or below 20. Through optimization, we discovered that look back lengths that fell beyond these bound-

aries did not generate acceptable expectations. An adaptive engine that generates useless values is useless in itself.

The Dynamic Break Out II initially looks back 20 days to determine its buy and sell levels. So when you start trading this system, your first buy point is the highest high of the past 20 days and your sell point is the lowest low of the past 20 days. At the end of each day, you measure the current market volatility by calculating the standard deviation of the past 30 day's closing prices. Market volatility can be measured using different calculations: average range, average true range, standard deviation of change in closing prices, and others. Once we determine today's market volatility, we compare it with yesterday's. If the volatility increases, then the number of look back days also increases. We change the number of look back days to the exact amount of the change in market volatility; if volatility increases by ten percent, then so does the number of look back days and vice versa.

The original Dynamic Break Out made its buying and selling decisions solely based on the highest high and lowest low values that were generated by our volatility-based adaptive engine. Once a position was initiated, a simple, yet effective, $1500 money management stop was put into place. The newer version uses the same entry technique in concert with an adaptive Bollinger Band. The length of the Bollinger Band calculation is the same number of look back days that is generated by the adaptive engine. The close of yesterday must be above the upper band and today's high must be greater than or equal to the highest high of *x* bars back before a long position can be initiated (*x* bars back is equal to our adaptive look back days value). Yesterday's close must be below the lower band and today's low must be less than or equal to the lowest low of *x* bars back before a short position can be taken. Instead of the simple money management stop, we incorporated a dynamic trailing stop. As we have discussed, the number of look back days changes on a daily basis. The adaptive engine decides the amount of change. The liquidation point of an existing trade is determined by calculating a simple moving average of closing prices for the past look back days. The sell liquidation would be just the opposite of the buy liquidation.

Dynamic Break Out II Pseudocode

```
If BarNumber = 1 then lookBackDays = 20
Else do the following
       Today's market volatility = StdDev(Close,30)
       Yesterday's market volatility = StdDev(Close[1],30)
       deltaVolatility = (today's volatility - yesterday's
          volatility)/today's volatility
       lookBackDays = (1 + deltaVolatility) * lookBackDays
       lookBackDays = MinList(lookBackDays,60)
       lookBackDays = MaxList(lookBackDays,20)
```

```
upBand = Average(Close,lookBackDays) + StdDev(Close,lookBackDays) *2.00
dnBand = Average(Close,lookBackDays) - StdDev(Close,lookBackDays) *2.00
buyPoint = Highest(High,lookBackDays)
sellPoint = Lowest(Low,lookBackDays)
longLiqPoint = Average(Close,lookBackDays)
shortLiqPoint = Average(Close,lookBackDays)
If Close of yesterday > upBand) then initiate a long position if today's
   market action >= buyPoint
If (Close of yesterday < dnBand) then initiate a short position if today's
   market action <= sellPoint
Liquidate long position if today's market action <= longLiqPoint
Liquidate short position if today's market action >= shortLiqPoint
```

Dynamic Break Out II Program

```
{Dynamic Break Out II by George Pruitt
This system is an extension of the original Dynamic Break Out system written
by George for Futures Magazine in 1996. In addition to the channel break out
methodology, DBS II incorporates Bollinger Bands to determine trade entry.}

Inputs: ceilingAmt(60),floorAmt(20),bolBandTrig(2.00);
Vars: lookBackDays(20),todayVolatility(0),yesterDayVolatility(0),
   deltaVolatility(0);
Vars: buyPoint(0),sellPoint(0),longLiqPoint(0),shortLiqPoint(0),upBand(0),
   dnBand(0);

todayVolatility = StandardDev(Close,30,1);
yesterDayVolatility = StandardDev(Close[1],30,1); {See how I offset the
                                                    function call to get
                                                    yesterday's value}
deltaVolatility = (todayVolatility - yesterDayVolatility)/todayVolatility;
lookBackDays = lookBackDays * (1 + deltaVolatility);
lookBackDays = Round(lookBackDays,0);
lookBackDays = MinList(lookBackDays,ceilingAmt); {Keep adaptive engine within
                                                  bounds}
lookBackDays = MaxList(lookBackDays,floorAmt);
upBand = BollingerBand(Close,lookBackDays,+bolBandTrig);
dnBand = BollingerBand(Close,lookBackDays,-bolBandTrig);

buyPoint = Highest(High,lookBackDays);
sellPoint = Lowest(Low,lookBackDays);

longLiqPoint = Average(Close,lookBackDays);
shortLiqPoint = Average(Close,lookBackDays);

if(Close > upBand) then Buy("DBS-2 Buy") tomorrow at buyPoint stop;
if(Close < dnBand) then SellShort("DBS-2 Sell") tomorrow at sellPoint stop;
```

```
if(MarketPosition = 1) then Sell("LongLiq") tomorrow at longLiqPoint stop;
if(MarketPosition = -1) then BuyToCover("ShortLiq") tomorrow at shortLiqPoint
   stop;
```

The Dynamic Break Out II program demonstrates how to:

- Measure market volatility by using the standard deviation of closing prices.
- Create a dynamic parameter using an adaptive engine

Dynamic Break Out II trading performance is summarized in Table 6.4.

A visual example of how this system enters and exits trades is shown in Figure 6.4.

Table 6.4
Dynamic Break Out II Performance

System Name: Dynamic Breakout Commission/Slippage = $75
Tested 1982–3/19/2002

Markets	Total Net Profit	Max. DrawDown	# of Trades	% Wins	Max. Cons. Losers
British Pound	$ 38,750.00	$ (43,612.50)	194	33.51%	20
Crude Oil	$ 21,237.50	$ (15,312.50)	109	35.78%	10
Corn	$ 3,050.00	$ (7,887.50)	120	34.17%	13
Copper	$ (25,175.00)	$ (25,862.50)	86	30.23%	7
Cotton	$ 25,555.00	$ (12,427.50)	112	33.93%	7
Deutsch Mark	$ 49,087.50	$ (7,837.50)	103	46.60%	5
Euro Currency	$ (7,062.50)	$ (10,950.00)	14	28.57%	4
Euro Dollar	$ 16,885.00	$ (5,025.00)	110	38.18%	8
Heating Oil	$ 30,728.10	$ (12,443.09)	113	39.82%	13
Japanese Yen	$ 118,200.00	$ (10,087.50)	98	51.02%	4
Live Cattle	$ (17,396.50)	$ (21,119.50)	125	25.60%	12
Natural Gas	$ 51,557.50	$ (14,902.50)	65	40.00%	7
Soybeans	$ (9,681.25)	$ (28,237.50)	128	33.59%	10
Swiss Franc	$ 57,337.50	$ (13,850.00)	106	47.17%	4
Treasury Note	$ 47,168.75	$ (6,646.88)	106	36.79%	7
U.S. Bonds	$ 67,093.75	$ (16,006.25)	107	40.19%	6
Wheat	$ (14,831.25)	$ (17,256.25)	124	31.45%	9
Total	$ 452,504.10		1820		

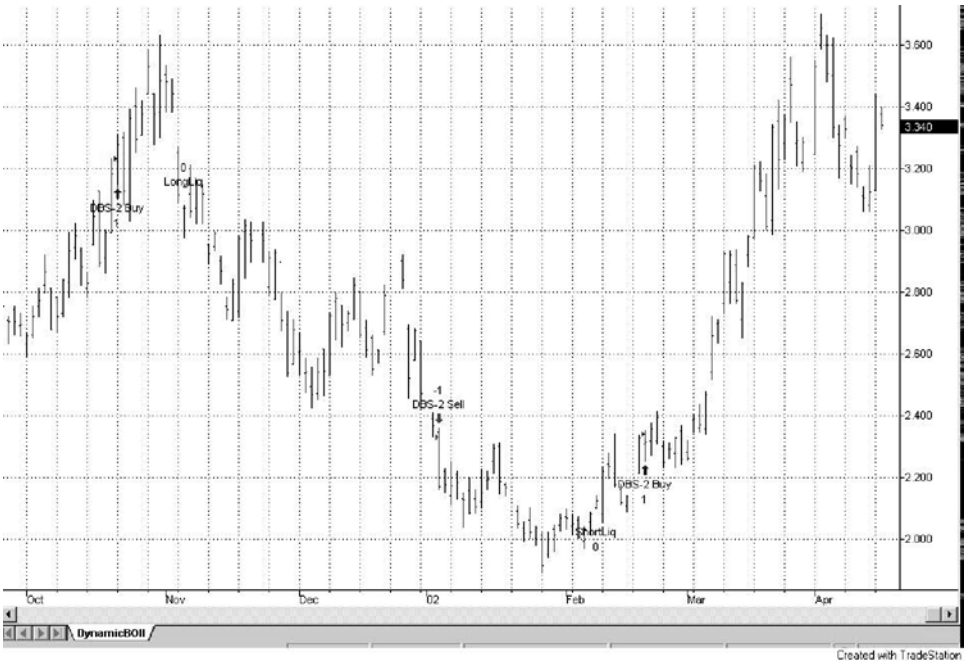

Figure 6.4 Dynamic Break Out II Trades

Dynamic Break Out II Summary

Yet again, another successful long-term trading approach. We guess we let the cat out of the bag . . .and what an ugly cat it is. The majority of successful trading systems are of the long-term trend following variety. Almost all traders realize this fact, but it doesn't stop them from searching out a shorter-term approach. See, the trend-following systems require diversification, which requires hefty capitalization. Also, trend-following systems can have substantial draw downs and go for years without making any money. The typical trader cannot persevere through these bad attributes, even though they know they will probably be rewarded in the long run.

Even this dynamic approach couldn't capture a profit in the soybean market. The continual failure of trend-following systems in the grain markets begs the question, "Why don't these systems work in the soybean or grain markets?" These markets move in a cyclical fashion due to the seasonality aspect of their underlying fundamentals. If we know ahead of time that these markets have this cyclical nature, then why can't we capture their movements? Cycles are very difficult to calculate and determine and, therefore, are usually overlooked. The two most predominant methods for finding cycles are trigonometric curve fitting and Fourier (spectral) analysis. The mathematics behind these two methods is relatively complex and detailed. We personally have

never seen a pure mathematically-based, cycle-finding trading system outperform the typical trend follower. If you do have any interest in this area, we refer you to John Ehlers, *Rocket Science for Traders* (John Wiley, 2001).

Before we move on, let's use our TradeStation for two different experiments. The first experiment will deal with the Dynamic Break Out II system and the soybean market. We saw how virtually useless the system was for capturing the trends in the soybean market. What would happen if we faded the trade signals? What we mean by fade is to do just the opposite. So, instead of buying at our long entry point, we will sell and vice versa. If the soybean market moves in cycles, which is countertrend, then we should be able to improve our performance by entering against the prevalent trend. Table 6.5 shows the performance of our countertrend soybean system.

No question that it did better, but overall it is still nothing to write home about. This somewhat proves that soybeans and other grain markets cannot be successfully traded by a longer-term trend-following approach. Since we are on the subject of cycles and seasonality, why don't we program a strategy that incorporates a seasonality filter? We will demonstrate how to use the keyword *date* to determine the current month and day. This system will trade the soybeans and will only take long signals from March 1 to July 1 and will only take

Table 6.5
Soybean Counter Trend Using Dynamic Break Out II

TradeStation Strategy Performance Report—DBSII Fade @S-Daily (6/17/82–4/11/02)			
Performance Summary: All Trades			
Total Net Profit	(1,681.25)	Open position P/L	0.00
Gross Profit	67,031.25	Gross Loss	(68,712.50)
Total # of trades	128	Percent profitable	64.06%
Number winning trades	82	Number losing trades	46
Largest winning trade	6,000.00	Largest losing trade	(11,012.50)
Average winning trade	817.45	Average losing trade	(1,493.75)
Ratio avg win/avg loss	.54725	Avg trade (win & loss)	(13.13)
Max consec. Winners	9	Max consec. losers	3
Avg # bars in winners	11	Avg # bars in losers	32
Max intraday drawdown	(29,043.75)		
Profit Factor	.97553	Max # contracts held	1
Account size required	29,043.75	Return on account	–5.79%

short signals from July 2 to February 28. These dates were derived from cyclical analysis of historical data on soybeans. (Table 6.6 shows the performance of our seasonal soybean system.)

```
Inputs: goLongStart(301),goLongEnd(701),goShortStart(702),goShortEnd(228);
Vars: monthAndDay(0);
{The inputs represent the months and days that we can enter long and short
trades}
{301 is March 01 >> can only go long from this date and up to
701 is July 01 >> this date
702 is July 02 >> can only go short from this date and up to
228 is February 28 >> this date}
{Let's use the date and extract the information that we need from it to
determine the month and the day}
{If we divide the date by 10000, the remainder is the month and day. We can
use the modulus function}
monthAndDay = Mod(Date of tomorrow,10000);
if(monthAndDay >= goLongStart and monthAndDay <= goLongEnd) then
begin
       buy("Seasonal Buy") tomorrow at Open;
end;
```

Table 6.6
Seasonal Soybean System Performance

TradeStation Strategy Performance Report—SeasonalSoybean @S-Daily (4/26/96–4/12/02)			
Performance Summary: All Trades			
Total Net Profit	(4362.50)	Open position P/L	1050.00
Gross Profit	13525.00	Gross Loss	(17887.50)
Total # of trades	11	Percent profitable	36.36%
Number winning trades	4	Number losing trades	7
Largest winning trade	7712.50	Largest losing trade	(5325.00)
Average winning trade	3381.25	Average losing trade	(2555.36)
Ratio avg win/avg loss	1.32320	Avg trade (win & loss)	(396.59)
Max consec. Winners	3	Max consec. losers	4
Avg # bars in winners	145	Avg # bars in losers	120
Max intraday drawdown	(15162.50)		
Profit Factor	.75611	Max # contracts held	1
Account size required	15162.50	Return on account	–28.77%

```
if(monthAndDay >= goShortStart or monthAndDay <= goShortEnd) then
{Notice that we had to use "or" instead of "and"—this is due
to the goShortEnd date is less than the goShortStart date}
begin
       sellShort("Seasonal Sell") tomorrow at Open;
end;
```