

The 3rd Order Polynomial Strategy Applied to British Pound Daily Future Prices Using Walk Forward, Out-Of-Sample Analysis.

Copyright © 2012 Dennis Meyers, Ph.D.

In a previous working paper entitled “The British Pound Cubed” we examined trading the British Pound (BP) using a Cubic Polynomial to serve as a better proxy for the BP market trend. The cubic polynomial should respond faster to changes in the underlying trend and model the trend changes more accurately than the straight line version. In that paper we used walk forward optimization with an in-sample period of five years of daily prices and an out-of-sample period of one year. We examined the following in-sample and out-of-sample periods:

In-Sample Period	Out-Of-Sample Period.
1/1/89 to 12/31/93	1/1/94 to 12/31/94
1/1/90 to 12/31/94	1/1/95 to 12/31/95
1/1/91 to 12/31/95	1/1/96 to 12/31/96
1/1/92 to 12/31/96	1/1/97 to 12/31/97
1/1/93 to 12/31/97	1/1/98 to 12/31/98

In that time period computers were not as fast nor had the memory as current computers. In addition the walk forward software was nowhere as sophisticated as current walk forward software. Given today’s technology we can bring the British Pound Cubed strategy up-to date and use shorter In-Sample and Out-Of-Sample periods so that constantly changing relationships in economic data and trading patterns in the currency world can be quickly mirrored in our in-sample data. As such we will examine an in-sample period of one year of BP daily prices followed by an out-of-sample (oos) period of three months of Daily BP prices. Each walk forward period will advance three months and we will examine 82 walk forward In-Sample and Out-Of-Sample periods from 1/1/1991 to 6/30/2012. **Table 1** below presents a table of the 82 In-Sample and Out-Of-Sample windows, the cubic strategy inputs and the quarterly out-of-sample profit/loss results using the performance metric filter described below.

The British Pound (BP) is a major currency traded world wide by corporations, institutions, banks, commodity funds and futures traders. The BP is traded 24 hours per day and most of the world’s largest banks make a two sided market in the BP and its associated derivatives. Small traders however, are constrained to trade the BP futures on the Chicago Mercantile Exchange (CME). The BP futures are traded from 7:20am to 2pm on the CME and from 2:30pm to 7:05am Monday through Thursday 5:30 P.M. to 7:05 AM Sundays and holidays on the CME Globex system. While the CME BP futures trading volume is small when compared to total world wide bank and institutional trading volume, arbitrage keeps the future prices in line with the bigger markets.

Data Discussion

The BP futures contract on the CME trades in the quarterly cycles of March, June, September and December. The current active BP futures contract is the DM Sep/2012. The British Pound is the currency of The United Kingdom. Each BP futures contract is worth the dollar value of

62,500 British Pounds. On May 29th, 1998 the BP Jun/98 closed at 1.6302 Dollars per BP making one BP contract worth \$101887.50 (1.6302 x 62,500). The BP future trades in units of \$0.0002 per BP. Thus a move of one tick of \$0.0002 is worth \$12.50 per contract ($\$0.0002 \$/BP \times 62,500 = \$12.50$).

The BP futures started trading on 1975. However, for this article we will limit our study to the price history from 1/1/91 to today. Here we will use a BP futures continuous contract. Since BP future contracts expire each quarter, a continuous contract is constructed by switching to the active contract on rollover day and back adjusting the difference in prices between the new contract and old contract thus creating a smooth continuous contract.

The performance results from strategies using continuous contracts cannot match actual results from trading real contracts because of the costs of actually having to rollover, and execution slippage. Execution slippage is the differences in prices from actually executing an order when a buy or sell signal is given and the price the computer system assumes that the order was executed at. When rolling over in real time via actual executions, the difference in prices between the new contract and the old contract on rollover day may not be the same as the closing prices the computer uses to construct the continuous contract.

Another problem with continuous contracts is that the past data is back adjusted by the difference in prices between the two contracts on rollover day. Systems developed on continuous contracts that use some form of percentage of prices will be impacted somewhat by this difference adjustment to past prices. Despite these qualifiers, system results based upon continuous contracts are still indicative of the worth of the trading system.

The Cubic Polynomial System.

In a previous working paper, entitled “The 2nd Order Polynomial Next Bar Forecast System on E-Mini 1min Bars”, we used the technique of least Squares to fit a 2nd order polynomial through 30+ closing prices to create a curve that served as a proxy for the market trend. When the curve moved up by a certain percent from its previous local low, the trend was assumed to have changed to the upside and a buy signal was given. When the curve moved down by a certain percent from its previous local high, the trend was assumed to have changed to the downside and a sell signal was given. Here I will use the least squares technique to fit a cubic polynomial of the form $a_0 + b_0 * t + c_0 * t^2 + d_0 * t^3$ to create a curve that will serve as a better proxy for the market trend. The cubic polynomial should respond faster to changes in the underlying trend and model the trend changes more accurately than the straight line version.

What is Least Squares?

Least squares is a mathematical technique where the squared vertical distance between the data and the curve that is being fit to the data is minimized. When the net squared distance (also called the sum of the squared errors) is minimized, a unique set of coefficients a_0, b_0, c_0 , and d_0 for the polynomial is determined.

For the cubic equation, the least squares coefficients are obtained from the solution of the following matrix equation.

$$\begin{vmatrix} T & \sum t & \sum t^2 & \sum t^3 \\ \sum t & \sum t^2 & \sum t^3 & \sum t^4 \\ \sum t^2 & \sum t^3 & \sum t^4 & \sum t^5 \\ \sum t^3 & \sum t^4 & \sum t^5 & \sum t^6 \end{vmatrix} \begin{vmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{vmatrix} = \begin{vmatrix} \sum p(t) \\ \sum (p(t) * t) \\ \sum (p(t) * t^2) \\ \sum (p(t) * t^3) \end{vmatrix}$$

where

$p(T)$ is today's price, $p(T-1)$ is yesterday's price and $p(1)$ is the price T days ago.

$\sum p(t)$ is the summation of prices from $t=1$ to T days

$\sum p(t) * t$ is the summation of prices times t from $t=1$ to T

$\sum t$ is the summation of the integer t from $t=1$ to T days

$\sum t^2$ is the summation of the integer squared from $t=1$ to T days
Etc.

Programs that solve for the solution to matrix equations can be found in the book "Numerical Recipes" by W. Press, et. al. which is currently available in book stores.

Once the coefficients to the cubic polynomial have been solved for we can generate the forecast for the next day's close which is given by the equation

$$P_f = a_0 + b_0 * (T+1) + c_0 * (T+1)^2 + d_0 * (T+1)^3$$

The Cubic Polynomial Strategy Defined

The least squares forecast is constructed by solving for the least squares coefficients each day using the last T days of closing prices. Then P_f is constructed from the equation above and plotted under the price chart. In general what we will be doing is following the plotted curve of P_f . When the curve increases by a percentage amount *pup* from the previous prior low of the curve we will go long. When the curve falls by the percentage amount *pdn* from the previous prior high of the curve we will go short. In addition we want to be able to jump aboard fast moving *major* changes in price direction. Thus if P_f is greater than P_f two days ago by the percentage amount *pjup* we will go long. If P_f is less than P_f two days ago by the percentage amount *pjdn* we will go short.

Buy Rule:

- IF P_f has moved up by more than the percentage amount of *pup* from the lowest low recorded in P_f while short then buy the BP futures at tomorrows open.

OR

- IF P_f is greater than P_f two days ago by the percentage amount of *pjup* then buy the BP futures at tomorrows open.

Sell Rule:

- IF P_f has moved down by more than the percentage amount p_{dn} from the highest high recorded in P_f while long then sell the BP futures at tomorrows open.

OR

- IF P_f is less than P_f two days ago by the percentage amount of p_{jdn} then sell the BP futures at tomorrows open.

Walk Forward Optimization

Walk forward optimization will be used here as a result of the nature of the British Pound prices. The BP vs Dollar relationship in general represents the political and economic differences between the economies of Great Britain and the U.S. as they relate to the world economies. This is a constantly changing relationship. As such, conditions that prevailed 1 year ago may no longer be representative of today's relationship.

To test this system we will use daily bar prices of British Pound(BP) futures contract traded on the CME and known by the symbol BP from January 1, 1991 to June 30, 2012

We will test this strategy with the above BP daily bars on a walk forward basis, as will be described below. To create our walk forward files we will use the **add-in** software product called the Power Walk Forward Optimizer (PWFO). In TradeStation (TS), we will run the PWFO strategy **add-in** along with the Cubic Polynomial Strategy on the BP Daily price data from 1/1/1991 to 6/30/2012. The PWFO will breakup and create 82 one year in-sample sections along with their corresponding 3 month(quarter) out-of-sample sections from the 21 years of BP daily prices in one TradeStation optimization run. (see Walk forward Testing below). The in-sample and out-of-sample section dates are shown in **Table 1** below. We will then use another software product called the Walk Forward Metric Explorer (WFME) on each of the 82 in-sample and out-of-sample(oos) sections generated by the PWFO to find the best in-sample section performance **filter** that determines the system input parameters (N , p_{up} , p_{dn} , p_{jup} , p_{jdn}) that will be used on the out-of-sample data. Detailed information about the PWFO and the WFME can be found at www.meyersanalytics.com

For the in-sample data we will run the TradeStation optimization engine on the 21 years of Bp daily bars with the following ranges for the nth order polynomial strategy inputs variables.

1. degree 3
2. T from 20 to 60 in steps of 4
3. p_{up} from 0.4 to 1.2 in steps of 0.2
4. p_{dn} from 0.4 to 1.2 in steps of 0.2
5. p_{jup} from 0.2 to 1 in steps of 0.2
6. p_{jdn} from 0.2 to 1 in steps of 0.2

This will produce 6875 different cases or combinations of the input parameters for each of the 82 PWFO in-sample and out-of-sample output files.

Walk Forward Out-of-Sample Testing

Walk forward analysis attempts to minimize the curve fitting or over fitting of price noise by using the law of averages from the Central Limit Theorem on the out-of-sample performance. In

walk forward analysis the data is broken up into many in-sample and out-of-sample sections. Usually for any strategy, one has some metric performance parameter selection procedure which we will call a filter used to select the strategy input parameters from the In-Sample optimization run. For instance, a *filter* might be all cases that have a profit factor (PF) greater than 1 and less than 3. For the number of cases left we might select the cases that had the best percent profit. This procedure would leave you with one case in the in-sample/test section output and its associated strategy input parameters. Now suppose we ran our optimization on each of our many in-sample/test sections and applied our filter to each in-sample/test section output. We would then use the strategy input parameters found by the filter in each in-sample/test section on the out-of-sample section immediately following that in-sample/test section. The input parameters found in each in-sample/test section and applied to each out-of-sample section would produce independent net profits and losses for each of the out-of-sample sections. Using this method we now have "x" number of independent out-of-sample section profits and losses from our filter. If we take the average of these out-of-sample section net profits and losses then we will have an estimate of how well our system will perform on average. Due to the Central Limit Theorem, as our sample size increases, the spurious noises results in the out-of-sample section performance should average out to zero in the limit leaving us with what to expect from our strategy and filter. Mathematical note: This assumption assumes that the out-of-sample returns are from probability distributions that have a finite variance.

Why use the walk forward technique? Why not just perform an optimization on the whole price series and choose the input parameters that give the best total net profits or profit factor? Surely the price noise cancels itself out with such a large number of test prices and trades. Unfortunately, nothing could be farther from the truth! Optimization is a misnomer and should really be called combinatorial search. As stated above, whenever we run a combinatorial search over many different combinations of input parameters or even with genetic(Grail) searches on noisy data on a fixed number of prices, *no matter how many*, the best performance parameters found are guaranteed to be due to “curve fitting” the noise and signal. What do we mean by “curve fitting”? Price series that we trade consists of random spurious price movements, which we call noise, and repeatable price patterns (*if they exist*). When we run, say, 8000 different input parameter combinations, the best performance parameters will be from those system input variables that are able to produce profits from the price pattern *and* the random spurious movements. While the price patterns will repeat, the same spurious price movements will not. If the spurious movements that were captured by a certain set of input parameters were a large part of the total net profits, as they are in real price data, then choosing these input parameters will produce losses when traded on future data. These losses occur because the spurious movements will not be repeated in the same way. This is why strategy optimization or combinatorial searches with no out-of-sample testing cause losses when traded in real time from something that looked great in the in-sample/test section. Unfortunately it is human nature to extrapolate past performance to project future trading results and so results from curve fitting give the illusion, a modern “siren call” so to speak, of future trading profits.

In order to gain confidence that our input parameter selection method using the optimization output of the test data will produce profits, we must test the input parameters we found on out-of-sample data. In addition, we must perform the in-sample/out-of-sample analysis many times.

Why not just do the out-of-sample analysis once or twice? Well just as in Poker or any card game, where there is considerable vagaries in hand to hand luck, walk forward out-of-sample analysis give considerable vagaries in quarter to quarter or quarter to quarter out-of-sample profit “luck”. That is, by pure chance we may have chosen some input parameter set that did well in in-sample/test section data *and* the out-of-sample section data. In order to minimize this type of “luck”, statistically, we must repeat the walk forward out-of-sample (oos) analysis over many (>30) in-sample/out-of-sample sections and take an average over all out-of-sample sections. This average gives us an expected quarterly return and a standard deviation of out-of-sample returns which allows us to statistically estimate the expected equity and its range for N out-of-sample periods in the future.

Finding The System Parameters Using Walk Forward Optimization

There are five system parameters to find T , pup , pdn , $pjup$, and $pjdn$.

For the test data we will run the TradeStation optimization engine on BP Daily price bars from 1/1/1991 to 6/30/2012 with the following ranges for the 3rd order polynomial strategy inputs variables. The Power Walk Forward Optimizer software will create 82 one year in-sample periods each followed by a 3 month out-of-sample period in this one TradeStation run(See table 1 for the in-sample/out-of-sample periods)

- 1 degree 3
- 2 T from 20 to 60 in steps of 4
- 3 pup from 0.4 to 1.2 in steps of 0.2
- 4 pdn from 0.4 to 1.2 in steps of 0.2
- 5 pjup from 0.2 to 1 in steps of 0.2
- 6 pdn from 0.2 to 1 in steps of 0.2

This will produce 6875 different cases or combinations of the input parameters for each of the 82 in-sample/out-of-sample files.

The question we are attempting to answer statistically, is which performance metric or combination of performance metrics (which we will call a *filter*) applied to the in-sample section will produce strategy inputs that produce statistically valid profits in the out-of-sample section. In other words we wish to find a performance metric *filter* that we can apply to the in-sample section that can give us strategy inputs that will produce, on average, good trading results in the future.

An example of a simple filter would be to choose the row in the in-sample section that had the highest Net Profit or perhaps a row that had the best Profit Factor. Unfortunately it was found that this type of simple filter very rarely produces good out-of-sample results. More complicated metric filters can produce good out-of-sample results minimizing spurious price movement biases in the selection of strategy inputs.

Here is an *example* of a better more complicated *filter* that was used in this paper. We require that the number of trades (NT) in the in-sample section be less than 36 trades a year. We require this so that we can minimize costs and stay with the major trend. Of course this is just the

author's preference for this strategy. After using a **NT** filter, as described, there can still be 100's of rows left in the in-sample section. There is a performance metric called standard deviation of trades in the in-sample section. We will call this metric *std*. Thus we would want the **std** to be as small as possible. Let us choose the **20** rows in the in-sample section that contain the **smallest (bottom) std** values from the rows that are left from the **NT** screen. In other words we sort **std** from low to high, eliminate the rows that have **NT**>36 and then choose the bottom 20 Rows of whatever is left. This particular filter will now leave **20** cases or rows in the in-sample file that satisfy the above filter conditions. We call this filter **b20std|nt36** where **b20std** means the bottom **20 std** rows left *after* the **NT** filter.. Suppose for this filter, within the 20 in-sample rows that are left, we want the row that has the minimum losing bars (**lbr**) metric in the in-sample/test section. The **metric lbr is the total losing bars from the losing trades in the in-sample section**. This would produce a filter which we will call **b20std|nt36-lbr**. For each **in-sample section** this **filter** leaves only one row in the in-sample/ section with its associated strategy inputs and out-of-sample net profit. This particular **b20std|nt36** is then applied to each of the 82 in-sample sections which give 82 sets of strategy inputs that are used to produce the corresponding 82 out-of-sample performance results. The average out-of-sample performance is calculated from these 82 out-of-sample performance results. In addition many other important out-of-sample performance statistics for this filter are calculated and summarized. **Figure 3** shows such a computer run along with a small sample of other filter combinations that are constructed in a similar manner. Row 3 of the sample output in Figure 3 shows the results of the filter discussed above.

Bootstrap Probability of Filter Results. Using modern "Bootstrap" techniques, we can calculate the probability of obtaining our filter's total out-of-sample *net* profits by chance. Here is how the bootstrap technique is applied. Suppose as an example, we have 100 files of in-sample/out-of-sample data .. A mirror filter is created 5000 times. However, instead of picking an out-of-sample net profit (OSNP) from a filter row as above, the mirror filter picks a **random** row's OSNP in each of the 100 PWFO files. Each of the 5000 mirror filters will choose a random row's OSNP of their own in each of the 100 PWFO files.. At the end, each of the 5000 mirror filters will have 100 **random** OSNP's picked from the rows of the 100 files. The sum of the 100 random OSNP picks for each mirror filter will generate a random total out-of-sample net profit (tOnpNet). The average and standard deviation of the 5000 mirror filter's different random tOnpNet's will allow us to calculate the chance probability of our filter's tOnpNet. Thus given the mirror filter's bootstrap random tOnpNet average and standard deviation, we can calculate the probability of obtaining our filter's tOnpNet by pure chance alone. Figure 3 lists the 5000 mirror filter's bootstrap average for our 82 out-of-sample files of \$25807 with a bootstrap standard deviation of \$30,514. The probability for obtaining our filters net profit of \$142,583 is 6.49×10^{-5} which is 3.83 standard deviations from the bootstrap average. For our filter, in row 3 in Figure 3, the expected number of cases that we could obtain by pure chance that would match or exceed the \$142583 is $961 \times 6.49 \times 10^{-5} = 0.0624$. where 961 is the total number of different filters we looked at in this run. This is much less than one so it is improbable that our result was due to pure chance.

Results

Table 1 below presents a table of the 82 quarters of in-sample and out-of-sample windows, the **Filter** selected, strategy inputs and the quarterly out-of-sample profit/loss results using the filter described above.

Table 2 below presents the yearly out-of-sample profit and losses computed from Table 1

Figure 1 presents a graph of the equity curve generated by using the filter on the 82 out-of-sample quarters ending 3/31/1992 – 6/30/2012. The equity curve is plotted from Equity column in Table 1. Plotted on the equity curve is the least squares straight line.

Figure 2 presents the out-of-sample BP daily bar chart from 10/18/2011 to 6/30/2012 with the cubic polynomial Indicator and all the buy and sell signals for those dates. For the cubic Polynomial Indicator, the black dots represent the cubic polynomial. The red line represents the percent down from its highest local high(pctdn) the cubic polynomial must decline to generate a sell signal and the blue line represents the percent up from its lowest local low(pctup) the cubic polynomial must rise to generate a buy signal.

Figure 3 Partial output of the Walk Forward Metric Performance Explorer

Discussion of System Performance

To see the effect of walk forward analysis, take a look at **Table 1**. Notice how the input parameters *degree*, *T*, *pup*, *pdn*, *pjup*, *pjdn* take sudden jumps from high to low and back. This is the walk forward process quickly adapting to changing volatility conditions in the In-Sample section. In addition, notice how often *pup* and *pdn* change from 0.4 to 1 and back. When the data gets very noisy with a lot of spurious price movements, the *pup* and *pdn*, has to be higher. During other times when the noise level is not as much *pup* and *pdn* can be lower to get onboard a trend faster.

In Figure 1, which presents a graph of the equity curve using the filter on the 82 quarters of out-of-sample data, notice how the equity curve follows the trend line with an R^2 of 0.976.

Using this filter, the strategy was able to generate \$142,583 net equity trading one BP contract for 82 quarters. The strategy generated an average of \$1739 per quarter. However, the strategy had a largest losing quarter of -\$8700 in the quarter ending in 9/30/1993 and largest winning quarter of \$21,969 in the quarter ending in 9/30/2008. The standard deviation of quarterly returns was \$4556 meaning that the quarterly profit will be between -\$2817 to \$6295 68% of the time. The largest number of losing quarters in a row was four for a 4 quarter loss of -\$11239 ending on 12/31/1993. The longest string of quarterly gains was eleven from 1/1/2002 to 9/30/2004. The longest time between new equity highs was 9 quarters.

In observing Table 1 we can see that this strategy and filter made trades from a low of 2 trades/quarter to a high of 13 trades/quarter with an average of 6.7 trades/quarter. The 13 trades/quarter occurred for the quarter ending 12/31/09.

In looking at Table2, the yearly out-of-sample results, the cubic strategy did quite well only showing losses in 4 out of 21 years. The cubic strategy did quite well in the financial meltdown of 2008 showing a profit of \$19,950 with only 28 trades that year..

Disclaimer

The strategies, methods and indicators presented here are given for educational purposes only and should not be construed as investment advice. Be aware that the profitable performance presented here is based upon hypothetical trading with the benefit of hindsight and can in no way be assumed nor can it be claimed that the strategy and methods presented here will be profitable in the future or that they will not result in losses.

References

1. Efron, B., Tibshirani, R.J., (1993), “An Introduction to the Bootstrap”, New York, Chapman & Hall/CRC.

**Table 1 Walk Forward Out-Of-Sample Performance Summary for
BP 3rd Order Polynomial Strategy**

BP Daily bars 1/1/1991 - 6/30/2012. The Strategy input values *degree(pw)*, *N*, *pup*, *pdn*, *pjup*, *pjdn* are the values found from applying the filter to the In-sample Section and applying the strategy inputs found to the out-of-sample section.

In-Sample Section Filter: #Trds \leq 36 and then bottom 20 std then minimum lbr

Where:

#Trds = The number of trades for a given set of strategy inputs in the test section.

std = The standard deviation of Trades in the in-sample section.

lbr = The total losing bars from the losing trades in the in-sample section

osnp = Quarterly Out-of-sample net profit in \$

Equity = running sum of the quarterly out-of-sample net profits in \$

ollt = The largest losing trade in the out-of-sample section in \$.

odd = The largest close drawdown in the out-of-sample section in \$.

ont = The number of trades in the out-of-sample quarter.

avosnp = The average profit per trade for that quarter \$

pw = polynomial degree. degree=3 for the cubic polynomial

T = is the number of bars lookback period to calculate the cubic polynomial.

pup = the percent up from its lowest local low the cubic polynomial must rise to generate a buy signal

pdn = the percent down from its highest local high the cubic polynomial must decline to generate a sell signal

pjup = The 2-day percentage jump up the cubic polynomial must rise to generate a buy signal.

pjdn = The 2-day percentage jump down the cubic polynomial must drop to generate a sell signal.

Qtr #	In-Sample Dates		Out-Of-Sample Dates		osnp	ollt	odd	ont	avosnp	NetEq	pw	T	pup	pdn	pjup	pjup
1	1/1/1991	to 12/31/1991	1/1/1992	to 3/31/1992	(5938)	-3525	-7263	9	(660)	(5938)	3	28	0.4	0.6	0.4	0.4
2	3/31/1991	to 3/30/1992	3/31/1992	to 6/30/1992	6525	-1513	-2200	6	1088	587	3	28	0.8	0.8	0.2	0.8
3	7/1/1991	to 6/30/1992	7/1/1992	to 9/30/1992	22538	-1750	-4438	6	3756	23125	3	20	0.4	1.2	0.4	1
4	10/2/1991	to 10/1/1992	10/2/1992	to 12/31/1992	6738	-4563	-7575	6	1123	29863	3	60	0.6	0.4	0.2	0.2
5	1/1/1992	to 12/31/1992	1/1/1993	to 3/31/1993	(1538)	-2438	-5250	11	(140)	28325	3	20	0.6	1.2	0.6	0.8
6	3/31/1992	to 3/30/1993	3/31/1993	to 6/30/1993	(513)	-2263	-2263	6	(86)	27812	3	60	0.4	0.4	0.2	0.2
7	7/1/1992	to 6/30/1993	7/1/1993	to 9/30/1993	(8700)	-3050	-9538	12	(725)	19112	3	28	0.6	0.4	0.4	0.4
8	10/2/1992	to 10/1/1993	10/2/1993	to 12/31/1993	(488)	-1225	-3625	7	(70)	18624	3	28	0.4	1.2	0.4	1
9	1/1/1993	to 12/31/1993	1/1/1994	to 3/31/1994	1513	-1000	-1000	4	378	20137	3	44	0.6	0.4	0.2	0.2
10	3/31/1993	to 3/30/1994	3/31/1994	to 6/30/1994	3238	-863	-1025	5	648	23375	3	28	0.4	1	0.4	0.8
11	7/1/1993	to 6/30/1994	7/1/1994	to 9/30/1994	(63)	-1038	-1038	2	(32)	23312	3	56	0.4	1	0.2	0.8
12	10/2/1993	to 10/1/1994	10/2/1994	to 12/31/1994	6000	0	0	2	3000	29312	3	56	0.4	1	0.2	0.8
13	1/1/1994	to 12/31/1994	1/1/1995	to 3/31/1995	(963)	-2038	-2688	9	(107)	28349	3	20	0.4	0.8	0.4	0.6
14	3/31/1994	to 3/30/1995	3/31/1995	to 6/30/1995	2525	-700	-1350	7	361	30874	3	20	0.4	0.6	0.4	0.4
15	7/1/1994	to 6/30/1995	7/1/1995	to 9/30/1995	5438	-613	-1088	7	777	36312	3	20	0.4	0.6	0.4	0.4
16	10/2/1994	to 10/1/1995	10/2/1995	to 12/31/1995	1250	-800	-950	9	139	37562	3	20	1	0.6	0.2	0.4
17	1/1/1995	to 12/31/1995	1/1/1996	to 3/31/1996	(2038)	-738	-1300	3	(679)	35524	3	60	0.6	0.8	0.6	0.4
18	3/31/1995	to 3/30/1996	3/31/1996	to 6/30/1996	(2125)	-1338	-2125	3	(708)	33399	3	44	0.8	0.8	0.8	0.8
19	7/1/1995	to 6/30/1996	7/1/1996	to 9/30/1996	(2013)	-1538	-2488	4	(503)	31386	3	56	0.6	0.4	0.6	0.4
20	10/2/1995	to 10/1/1996	10/2/1996	to 12/31/1996	3138	-1688	-1688	10	314	34524	3	20	0.6	0.4	0.2	0.4
21	1/1/1996	to 12/31/1996	1/1/1997	to 3/31/1997	6038	-1325	-1475	6	1006	40562	3	24	0.6	0.4	0.6	0.2
22	3/31/1996	to 3/30/1997	3/31/1997	to 6/30/1997	425	-738	-2200	7	61	40987	3	24	0.4	0.4	0.2	0.2
23	7/1/1996	to 6/30/1997	7/1/1997	to 9/30/1997	2600	-1738	-2450	10	260	43587	3	24	0.4	0.4	0.2	0.2
24	10/2/1996	to 10/1/1997	10/2/1997	to 12/31/1997	4225	-1125	-2563	8	528	47812	3	24	0.4	0.4	0.2	0.4
25	1/1/1997	to 12/31/1997	1/1/1998	to 3/31/1998	1638	-913	-2538	8	205	49450	3	20	0.4	1.2	0.4	0.6
26	3/31/1997	to 3/30/1998	3/31/1998	to 6/30/1998	3900	-1113	-1113	8	488	53350	3	20	0.4	0.8	0.4	0.6
27	7/1/1997	to 6/30/1998	7/1/1998	to 9/30/1998	5500	-820	-820	5	1100	58850	3	28	1	0.4	0.6	0.2
28	10/2/1997	to 10/1/1998	10/2/1998	to 12/31/1998	(1700)	-1775	-2800	9	(189)	57150	3	28	0.6	0.4	0.2	0.2
29	1/1/1998	to 12/31/1998	1/1/1999	to 3/31/1999	2225	-575	-575	6	371	59375	3	28	1.2	0.4	0.6	0.2
30	3/31/1998	to 3/30/1999	3/31/1999	to 6/30/1999	(2650)	-1750	-3263	10	(265)	56725	3	24	0.4	0.4	0.4	0.2
31	7/1/1998	to 6/30/1999	7/1/1999	to 9/30/1999	5950	0	0	3	1983	62675	3	48	0.4	1	0.2	0.8
32	10/2/1998	to 10/1/1999	10/2/1999	to 12/31/1999	(613)	-1163	-1163	3	(204)	62062	3	48	1	1	0.4	0.6
33	1/1/1999	to 12/31/1999	1/1/2000	to 3/31/2000	(288)	-1875	-2325	5	(58)	61774	3	36	1.2	1	1	0.6
34	3/31/1999	to 3/30/2000	3/31/2000	to 6/30/2000	3488	-1225	-1225	3	1163	65262	3	56	0.6	0.8	0.6	0.4
35	7/1/1999	to 6/30/2000	7/1/2000	to 9/30/2000	2475	-2913	-2913	8	309	67737	3	20	0.4	1	0.2	1
36	10/2/1999	to 10/1/2000	10/2/2000	to 12/31/2000	(2013)	-1750	-7075	8	(252)	65724	3	20	0.6	1	0.2	1
37	1/1/2000	to 12/31/2000	1/1/2001	to 3/31/2001	1538	-863	-863	7	220	67262	3	20	0.8	1	0.2	1
38	3/31/2000	to 3/30/2001	3/31/2001	to 6/30/2001	(2600)	-1750	-5488	9	(289)	64662	3	20	1	0.6	0.6	0.4
39	7/1/2000	to 6/30/2001	7/1/2001	to 9/30/2001	(963)	-2275	-2275	2	(482)	63699	3	60	1.2	0.8	1	0.6
40	10/2/2000	to 10/1/2001	10/2/2001	to 12/31/2001	(4338)	-1900	-5238	10	(434)	59361	3	20	1.2	0.6	0.8	0.6
41	1/1/2001	to 12/31/2001	1/1/2002	to 3/31/2002	820	-413	-650	3	350	60411	3	36	1	0.4	1	0.2
42	3/31/2001	to 3/30/2002	3/31/2002	to 6/30/2002	4738	-925	-925	2	2369	65149	3	40	1	1	1	1
43	7/1/2001	to 6/30/2002	7/1/2002	to 9/30/2002	4288	-400	-400	4	1072	69437	3	40	0.4	0.4	0.4	0.2
44	10/2/2001	to 10/1/2002	10/2/2002	to 12/31/2002	1475	0	0	5	295	70912	3	32	0.6	0.4	0.4	0.2
45	1/1/2002	to 12/31/2002	1/1/2003	to 3/31/2003	6663	-1125	-1613	6	1111	77575	3	24	0.6	0.4	0.4	0.2
46	3/31/2002	to 3/30/2003	3/31/2003	to 6/30/2003	820	-1225	-1713	5	210	78625	3	24	1.2	1.2	0.8	0.8
47	7/1/2002	to 6/30/2003	7/1/2003	to 9/30/2003	2825	-1950	-2463	7	404	81450	3	24	1.2	0.6	0.8	0.4
48	10/2/2002	to 10/1/2003	10/2/2003	to 12/31/2003	5906	-1631	-1631	6	984	87356	3	20	0.8	0.8	0.6	0.6
49	1/1/2003	to 12/31/2003	1/1/2004	to 3/31/2004	1631	-1888	-3400	11	148	88987	3	20	1.2	0.8	0.6	0.2
50	3/31/2003	to 3/30/2004	3/31/2004	to 6/30/2004	569	-1363	-6681	12	47	89556	3	20	1	0.8	0.8	0.2
51	7/1/2003	to 6/30/2004	7/1/2004	to 9/30/2004	3263	-1631	-2338	7	466	92819	3	20	1.2	0.6	0.8	0.2
52	10/2/2003	to 10/1/2004	10/2/2004	to 12/31/2004	(625)	-1919	-3906	9	(69)	92194	3	20	1.2	0.4	0.8	0.2
53	1/1/2004	to 12/31/2004	1/1/2005	to 3/31/2005	2094	-2544	-3981	4	524	94288	3	40	0.6	0.4	0.6	0.2
54	3/31/2004	to 3/30/2005	3/31/2005	to 6/30/2005	4156	-1406	-3744	6	693	98444	3	28	1.2	0.8	1	0.6
55	7/1/2004	to 6/30/2005	7/1/2005	to 9/30/2005	(1100)	-2700	-3781	8	(138)	97344	3	20	0.8	0.8	0.6	0.8
56	10/2/2004	to 10/1/2005	10/2/2005	to 12/31/2005	2188	-1794	-4219	6	365	99532	3	24	0.6	0.6	0.6	0.2

Qtr #	In-Sample Dates		Out-Of-Sample Dates		osnp	ollt	odd	ont	avosnp	NetEq	pw	N	pup	pdn	pjup	pjup
57	1/1/2005	to 12/31/2005	1/1/2006	to 3/31/2006	1931	-1319	-2444	8	241	101463	3	20	0.6	0.8	0.6	0.8
58	3/31/2005	to 3/30/2006	3/31/2006	to 6/30/2006	(2388)	-2500	-5613	8	(299)	99075	3	20	0.4	1	0.4	1
59	7/1/2005	to 6/30/2006	7/1/2006	to 9/30/2006	4675	-1231	-1669	8	584	103750	3	20	1.2	0.4	0.8	0.2
60	10/2/2005	to 10/1/2006	10/2/2006	to 12/31/2006	2375	-1075	-1075	8	297	106125	3	20	1.2	0.6	0.8	0.2
61	1/1/2006	to 12/31/2006	1/1/2007	to 3/31/2007	(81)	-1306	-2213	7	(12)	106044	3	28	0.8	0.4	0.6	0.2
62	3/31/2006	to 3/30/2007	3/31/2007	to 6/30/2007	1313	-994	-1700	5	263	107357	3	28	0.8	0.4	0.6	0.2
63	7/1/2006	to 6/30/2007	7/1/2007	to 9/30/2007	1213	-1563	-1563	8	152	108570	3	24	1	0.4	1	0.2
64	10/2/2006	to 10/1/2007	10/2/2007	to 12/31/2007	6538	-888	-888	5	1308	115108	3	24	1.2	0.4	1	0.2
65	1/1/2007	to 12/31/2007	1/1/2008	to 3/31/2008	(3294)	-1381	-2150	7	(471)	111814	3	24	1.2	0.6	1	0.4
66	3/31/2007	to 3/30/2008	3/31/2008	to 6/30/2008	1019	-2325	-4231	6	170	112833	3	20	1.2	0.8	1	0.8
67	7/1/2007	to 6/30/2008	7/1/2008	to 9/30/2008	21969	-1369	-1369	7	3138	134802	3	20	1.2	0.6	1	0.4
68	10/2/2007	to 10/1/2008	10/2/2008	to 12/31/2008	256	-5888	-10794	8	32	135058	3	36	0.4	1	0.2	0.4
69	1/1/2008	to 12/31/2008	1/1/2009	to 3/31/2009	(94)	-2469	-3513	10	(9)	134964	3	28	1.2	0.4	0.4	0.4
70	3/31/2008	to 3/30/2009	3/31/2009	to 6/30/2009	(3356)	-3719	-7275	9	(373)	131608	3	40	0.6	0.4	0.2	0.2
71	7/1/2008	to 6/30/2009	7/1/2009	to 9/30/2009	826	-3194	-4813	7	151	132664	3	28	1.2	0.4	0.8	0.4
72	10/2/2008	to 10/1/2009	10/2/2009	to 12/31/2009	(6125)	-2894	-7238	13	(471)	126539	3	28	0.4	0.6	0.2	0.2
73	1/1/2009	to 12/31/2009	1/1/2010	to 3/31/2010	3200	-1525	-1525	6	533	129739	3	28	0.4	1	0.2	0.4
74	3/31/2009	to 3/30/2010	3/31/2010	to 6/30/2010	2875	-1969	-2200	7	411	132614	3	24	0.8	0.8	0.8	0.2
75	7/1/2009	to 6/30/2010	7/1/2010	to 9/30/2010	5069	-1275	-2106	6	845	137683	3	24	0.4	1	0.4	1
76	10/2/2009	to 10/1/2010	10/2/2010	to 12/31/2010	(3338)	-2750	-5731	8	(417)	134345	3	24	0.4	1.2	0.4	0.8
77	1/1/2010	to 12/31/2010	1/1/2011	to 3/31/2011	(1006)	-1975	-2088	5	(201)	133339	3	24	0.4	1.2	0.4	0.8
78	3/31/2010	to 3/30/2011	3/31/2011	to 6/30/2011	1031	-1538	-1656	7	147	134370	3	24	0.6	1.2	0.2	0.8
79	7/1/2010	to 6/30/2011	7/1/2011	to 9/30/2011	3413	-1044	-2056	5	683	137783	3	44	0.4	0.4	0.2	0.2
80	10/2/2010	to 10/1/2011	10/2/2011	to 12/31/2011	981	-1363	-3656	6	164	138764	3	48	1	0.4	0.8	0.2
81	1/1/2011	to 12/31/2011	1/1/2012	to 3/31/2012	(1250)	-1181	-2563	8	(156)	137514	3	20	1	1	0.6	0.8
82	3/31/2011	to 3/30/2012	3/31/2012	to 6/30/2012	5069	-281	-281	5	1014	142583	3	28	0.6	0.4	0.4	0.2

Table 2 Walk Forward Yearly Out-Of-Sample Performance Summary for the BP 3rd Order Polynomial Strategy

Date	YearlyP/L	Yearly NT
12/31/1992	\$29,863	27
12/31/1993	(\$11,239)	36
12/31/1994	\$10,688	13
12/31/1995	\$8,250	32
12/31/1996	(\$3,038)	20
12/31/1997	\$13,288	31
12/31/1998	\$9,338	30
12/31/1999	\$4,912	22
12/31/2000	\$3,662	24
12/31/2001	(\$6,363)	28
12/31/2002	\$11,551	14
12/31/2003	\$16,444	24
12/31/2004	\$4,838	39
12/31/2005	\$7,338	24
12/31/2006	\$6,593	32
12/31/2007	\$8,983	25
12/31/2008	\$19,950	28
12/31/2009	(\$8,519)	39
12/31/2010	\$7,806	27
12/31/2011	\$4,419	23
6/30/2012	\$3,819	13
Average	\$6,790	26

Figure 1 Graph of Strategy Out-Of Sample Equity Applying the Walk Forward Filter Each Quarter On BP Daily Prices 1/1/1992 – 6/30/2012

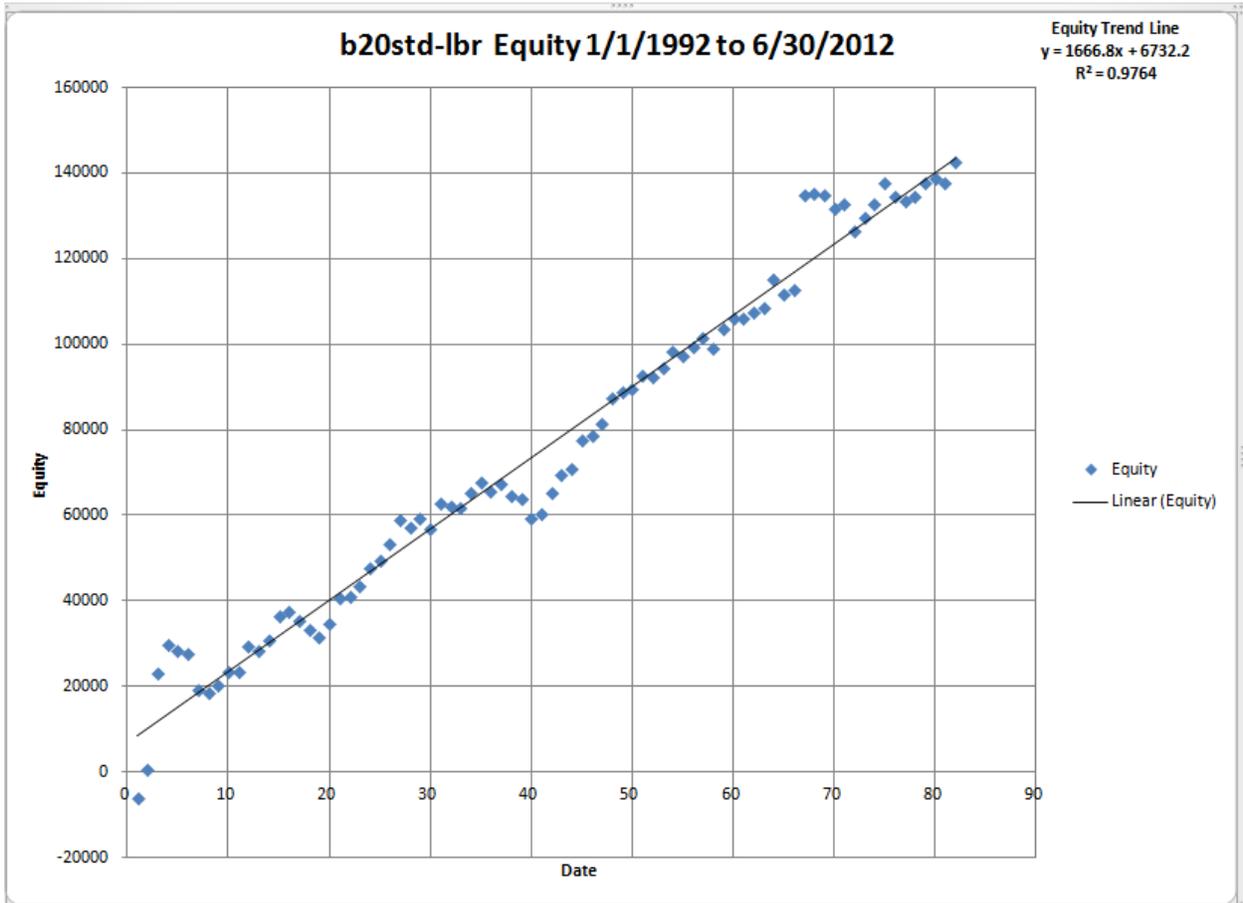
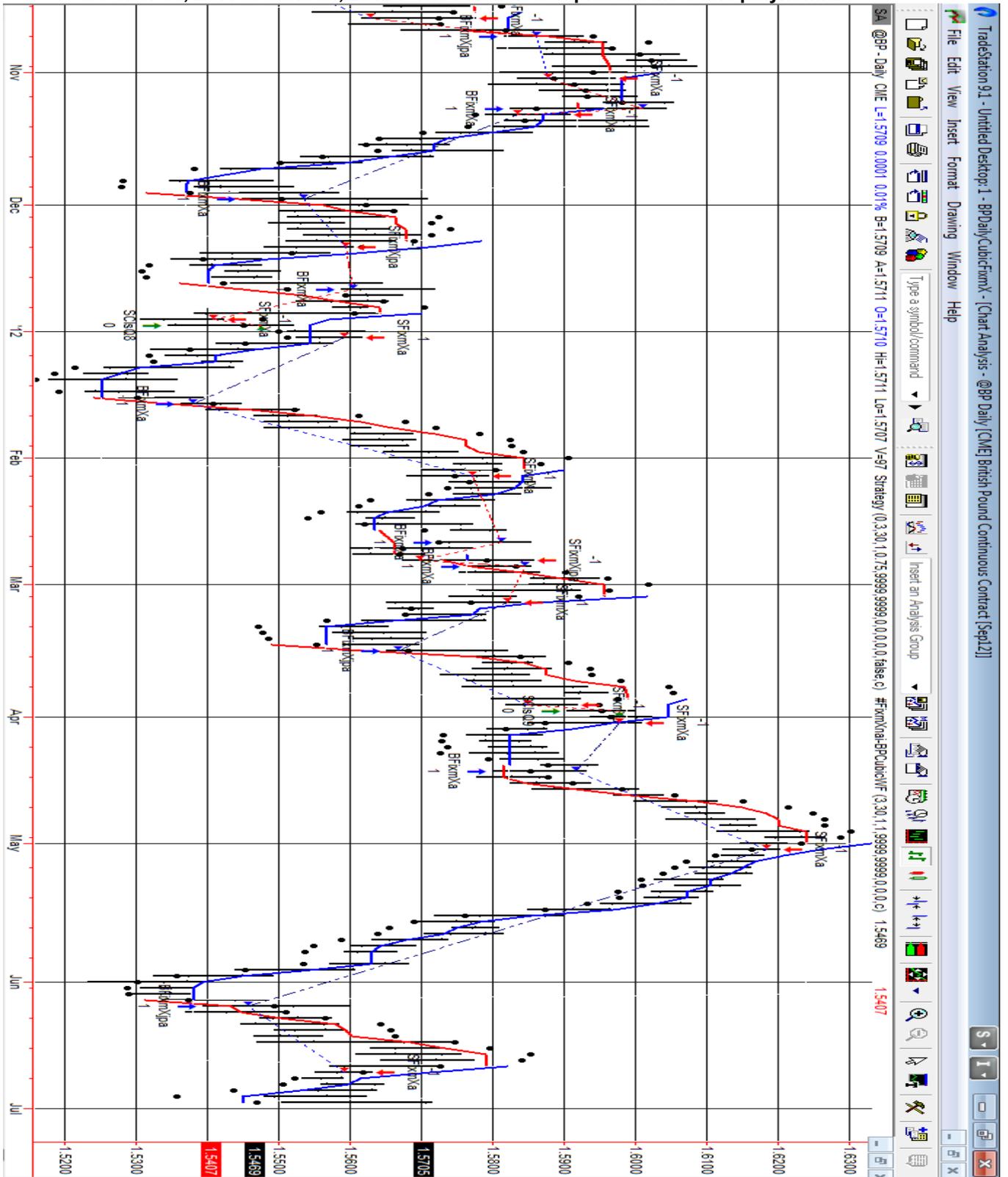


Figure 2 BP Daily Prices with Walk Forward Filter Inputs to the 3rd Order Polynomial
 October 18, 2011 to June 30, 2012 . The black dots represent the cubic polynomial



**Figure 3 Partial output of the Walk Forward Metric Performance Explorer
BP Daily, 3rd Order Polynomial Strategy**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1	BPdCubicPoly1y3m	19920331	20120631	#82	nT<=36	Boot Random Ave=25807				Boot Random Std=30514				f961							
2	Filter-Metric	tOnp	mOnp	aOnp	aOTr	aOnT	%P	t	std	llQtr	eqDD	lr	#	eqTrn	eqR2	Blw	BE	ndd	tOnpNet	Prob	
3	b20std-lbr	142583	1394	1739	258.8	6.7	0.7	3.46	4556.6	-8700	-11239	4	82	1667	98	9	27.5	7.9	142583	6.49E-05	
4	b20std-mtnp	135271	1526	1650	242	6.8	0.7	3.29	4545.4	-6575	-10676	4	82	1620	96	11	30.4	7.9	135271	1.67E-04	
5	b20std-wbr	130071	1494	1586	233.9	6.8	0.7	3.02	4759.8	-12225	-20001	4	82	1546	96	18	36	15.4	130071	3.17E-04	

The Filter Output Columns are defined as follows:

Row 1 BPdCubicPoly1y3m is the Out-Of-Sample(OOS) File name, First OOS Qtr Date(19920331), Last OOS Qtr Date(20120631), **Number of OOS Quarters=#82**, **NT<=36** =In-Sample Number of Trades <=36, **Boot Random Ave**=average of bootstrap random picks. **Boot Random Std**= standard deviation of bootstrap random picks. **F961** =number of different filters examined.

Filter-Metric = the filter that was run. Row 3 filter-metric chosen is **b20std-lbr**

The **b20std-lbr** filter produced the following average **82 OOS Quarter** statistics on row 3.

tOnp = Total out-of-sample(oos) net profit for these 82 quarters.

mOsp = median oos net profit for the 82 quarters

aOsp = Average oos net profit for the 82 quarters

aanp = Average oos profit per trade

aOnT = Average number of oos trades per quarter

%P = The percentage of oos quarters that were profitable

t = The student t statistic for the 82 quarterly oos profits. The higher the t statistic the higher the probability that this result was not due to pure chance

std = The standard deviation of the 82 quarterly oos profits

llw = The largest losing oos quarter

eqDD = The oos equity drawdown

lr = The largest number of losing oos quarters in a row

= The number of quarters this filter produced a quarterly result. Note for some quarters there can be no strategy inputs that satisfy a given filter's criteria.

b00 = The straight line trend of the oos gross profit equity curve in \$/quarter.

Blw = The maximum number of quarters the oos equity curve failed to make a new high.

BE = Break even quarters. Assuming the average and standard deviation are from a normal distribution, this is the number of quarters you would have to trade to have a 99% probability that your oos equity is above zero.

ndd = The normalized equity drawdown = $100 \cdot \text{eqDD} / \text{tONet}$

tOnpNet = Total out-of-sample net profit(tOnpNet) minus the total trade cost.
 $\text{tOnpNet} = \text{tOnp} - \# \cdot \text{aOnT} \cdot \text{Cost}$. Note Cost was zero.

Prob = probability that the filter's tOnpNet was due to pure chance.