

Trading Crude Light 5min Bars Using The End Point Fast Fourier Transform

Working Paper June 2010
Copyright © 2010 Dennis Meyers

In the working paper entitled “The Discrete Fourier Transform Illusion” , www.meyersanalytics.com/publications/dft.pdf, we demonstrated the misuses of the mathematical technique called the Fourier Transform as applied to the S&P500 index. There we showed how fitting the Fourier Transform to the S&P500 index data series produced a perfect curve fit on past data giving the illusion that this technique would predict the major turning points of the S&P500 index. Why does this happen? When the FFT went to fit the data, it already knew where all the tops and bottoms were. The FFT mathematics minimizes the error between the curve it generates and the real data points. This error minimization process forces the generated curve to smoothly fit the past data like a glove. As a matter of fact, it’s almost impossible not to get an excellent fit. However, as we demonstrated, when we examined the Fourier Transform on a day-by-day walk forward basis this seemingly wondrous predictive capability disappeared.

Here it will be shown how to use EPFFT technique on a walk forward basis to trade 5-minute bars of Crude Light (CL) futures.

The End Point FFT

In order to avoid the past data curve fit illusion, we will create an indicator that walks forward one bar at a time. This indicator will calculate the noise filtered FFT curve but only save the last point, or end point, of the curve on the day that it is calculated. We will then connect all the generated end points to produce a curve that matches what we would have seen if we performed the noise filtered FFT on the end point dates.

EPFFT Construction Details

Unfortunately constructing the noise filtered FFT of a price data series is not quite as simple as just taking 512 closing prices, and directly plugging them into a FFT algorithm.

The mathematics of the Discrete Fourier Transform(DFT) assumes that the time-domain sample is periodic and that it has captured an integral number of periods. This means that the DFT assumes the end of the sampled series implicitly wraps around to the beginning to start all over again. Thus for 512 sampled data points the DFT assumes that these 512 sampled data points repeat every 512 samples. With real data series this is seldom the case and this creates what is called a wraparound effect in the frequency domain. The wraparound effect creates a lot of distortion on the ends of the price series when the noise filtered FFT is transformed back. Unfortunately the end point of the price series is the very point we wish to estimate. While we can do nothing about the wrap around violation, we can significantly lessen it’s effects by what is called zero padding.

The DFT assumes the time domain sample is periodic and repeats. Suppose a price series starts at 400 and wiggles and wags for 512 data samples ending at the value of 600. The DFT assumes that the price series starts at zero, suddenly jumps to 400, goes to 600 and suddenly jumps down to zero again and then repeats. The DFT must create all kinds of different frequencies in the frequency domain to try and match this type of behavior. These false frequencies created to match the jumps and the high average price completely swamp the amplitudes of any real frequencies making them look like noise. Fortunately this effect can be almost eliminated by a simple technique called end point flattening.

The calculation of end point flattening coefficients is simple. If $x(1)$ represents the first price in the sampled data series, $x(n)$ represent the last point in the data series and $y(i)$ equal to the new endpoint flattened series then:

$$a = x(1) \quad b=(x(n)-x(1))/(n-1)$$

$$y(i) = x(i) - [a+ b*(i-1)] \quad \text{for } i=1 \text{ to } n \quad (1)$$

We can see that when $i=1$ then $y(1)=0$ and when $i=n$ then $y(n)=0$. What we’ve done is subtract the beginning value of the time series to make the first value equal to zero and then rotate the rest of the time series such that the end

point is now zero. This technique reduces the endpoint distortion but introduces a low frequency artifact into the Fourier Frequency spectrum.

EPFFT Curve Construction

For this article a sliding time-bar window of 512 5-minute bars of CL continuous futures from 5/20/08 to 6/18/10 will be used.

Step 1 Take the Log_e of the 512 prices to minimize the exponential trend movements of the E-mini. End flatten these 512 log prices using equation (1) above.

Step 2 Take the FFT of these 512 points to create 512 frequency domain complex numbers f_i .

Step 3 Use a Low Pass Threshold filter that zeros out the frequencies whose magnitudes are less than $\text{thres} * F_{\text{max}}$. Thres is the decimal percentage of F_{max} and F_{max} is the maximum frequency amplitude of the spectrum.

Step 4 Do an inverse FFT on the noise filtered spectrum and save the last 2 points. Although it's not obvious yet, we will need the noise filtered point one sample before the endpoint.

Add back the end point flattening of **Step 1** and taking the exponential value to reverse the log_e

Step 5 Save the filtered end point and the point before. Call the endpoint $ep(k)$ and the point before $ep1(k)$ where k is the order of the sliding window. That is, the first sliding window $k=1$, the second, $k=2$, etc. Slide the 512 bar data window forward one bar, and repeat steps 1 through 4.

When the data window is moved forward one bar at a time a new data sample is added to the end and the data sample at the beginning is subtracted. This adding and subtracting causes the end point flattening coefficients and the power in the frequency spectrum to jump around creating distortion and jitter in the calculation of the noise filtered end point. This random jumping of the FFT endpoints as the data window slides forward in time adds a random jump to FFT end point curve. Fortunately this jumping can be minimized by creating a new end point curve from the two saved noise filtered end points, $ep(k)$ and $ep1(k)$, above in step 5. Since turning points are of interest rather than magnitude then in **step 5** a new variable will be created called **sumEP** where

$$\text{sumEP}(k) = \text{sumEP}(k-1) + ep(k) - ep1(k)$$

This new curve **sumEP(k)** is the sum of all the changes in the individual $ep(k)$'s from their noise filtered FFT value one sample before. This change series minimizes the magnitude jump problem creating a fairly smooth EPFFT curve.

The EPFFT System Defined

Even though **sumEP** is a fairly smooth curve it still has a number of short term wiggles preventing us from simply going long when the curve turns up and going short when the curve turns down. To create a system, we will use a simple curve following technique on CL 5-minute bars.

Buy Rule:

- **IF sumEP** has moved up by more than the point amount of $pntup$ from the lowest low recorded in **sumEP** while short then buy the CL futures at the market..

Sell Rule:

- **IF sumEP** has moved down by more than the point amount $pntdn$ from the highest high recorded in **sumEP** while long then sell the CL futures at the market.

Intraday Bars Exit Rule:

Close the position at 1420 ET before the CL open outcry close at 1430 ET (no trades will be carried overnight).

Intraday Bars First Trade of Day Entry Rule:

Ignore all trade signals before **8:30am ET**. For the Buy and Sell rules above we have included a first trade of the day entry rule. Trading in the CL futures has changed a lot in the last 5 years because of 24hr Globex trading. In

particular trading starts a lot earlier in the morning when Asia and then Europe opens and then dies down. The CL volume starts to pick up again around 830am ET before the open outcry floor opens at 9am ET so we will have the strategy resume trading at 830am ET

Discussion of Crude Light Future Prices

Crude Light futures (CL) are traded on Globex and on the trading floor at the NYMEX. On Globex the CL is traded on a 23hour basis . The NYMEX hours for open outcry floor trading (RTH) are 9:00 ET to 14:30 ET. Over 60% of the volume in the CL is done on Globex during the NYMEX open outcry hours. We have restricted our study to only trading the CL during the 8:30 to 14:20 ET hours.

Testing The EPFFT System Using Walk Forward Optimization

There will be three strategy parameters to determine:

1. *thres*, Threshold filter.
2. *pntrup*, if sumEP has moved up by more than the point amount of *pntrup* from the lowest low recorded in sumEP while short then issue a buy signal
3. *pntrdn*, if sumEP has moved down by more than the point amount *pntrdn* from the highest high recorded in sumEP while long then sell

To test this system we will use five-minute bar prices of Crude Light (CL) futures contract traded on the NYMEX/Globex and known by the symbol CL for the 104 weeks from June 27, 2008 to June 18, 2010.

We will test this strategy with the above CL 5 min 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 End Point Fast Fourier Transform Strategy on the CL 5min bar data from June 27, 2008 to June 18, 2010..

What Is A Test Section and Out-Of-Sample Section?

Whenever we do a TS optimization on a number of different strategy inputs, TS generates a report of performance metrics (total net profits, number of losing trades, etc) vs these different inputs. If the report were sorted on say the total net profits(*tnp*) performance metric column then the highest *tnp* would correspond to a certain set of inputs. This is called a *test section*. If we choose a set of strategy inputs from this report based upon some performance metric we have no idea whether these strategy inputs will produce the same results on future price data or data they have not been tested on. Price data that is not in the test section is defined as *out-of-sample data*. Since the performance metrics generated in the test section are mostly due to “curve fitting” (see Walk Forward Out-of-Sample Testing section below) it is important to see how the strategy inputs chosen from the test section perform on out-of-sample data.

What Does The Power Walk Forward Optimizer (PWFO) Do?

The PWFO is a TS *add-in* that breaks up the TS optimization run into a number of user selectable test and out-of-sample sections. The PWFO prints out the test sample performance **and the out-of-sample performance results**, on one line, for each case or input variable combination that is run by the TradeStation(TS) optimization module to a user selected spreadsheet comma delimited file. The PWFO can generate up to 500 different test and out-of-sample date optimization files in one TS run, saving the user from having to generate optimization runs one at a time. The PWFO output allows you to quickly determine whether your procedure for selecting input parameters for your strategy just curve fits the price and noise, or produces statistically valid out-of-sample results. In addition to the out-of-sample performance results presented for each case, 30+ superior and robust performance metrics (many are new and never presented before) are added to each case line in the test section and printed out to the comma delimited file. These 30+ performance metrics allow for a superior and robust selection of input variables from the test section that have a higher probability of performing well on out-of-sample data (Please see Appendix 2 for a listing of these performance metrics).

For our computer run we will have the PWFO breakup the 104 weeks of CL five-minute bar price data into 104 test/out-of sample files. The test sections will be 30 calendar days and the out-of-sample(oos) section will be the one week following the test section. The oos week will always end on a Friday as will the 30-day calendar test section. As an example the first test section would be from 5/22/2008 to 6/20/2008 and the out-of-sample section would be from 6/23/2008 to 6/27/2008.(our test and out-of-sample sections always end on a Friday). We would

then move everything ahead a week and the 2nd test section would be from 5/29/2008 to 6/27/2008 and the out-of-sample section would be from 6/30/2008 to 7/04/2008. Etc.

The PWFO 104 test/out-of-sample section dates are shown in **Table 1** on page 8 below. We will then use another software product called the Walk Forward Performance Metric Explorer (WFPME) on each of the 104 test and out-of-sample(oos) sections generated by the PWFO to find the best test section performance *filter* that determines the system input parameters (*thres, pntup, pntdn*) that will be used on the out-of-sample data. Detailed information about the PWFO and the WFPME can be found at www.meyersanalytics.com

For the test data we will run the TradeStation optimization engine on the 104 weeks of CL 5 min bars with the following ranges for the EPFFT strategy input variables.

1. thres from 4 to 20 in steps of 4
2. pntup from 0.5 to 5 steps of 0.25
3. pntdn from 0.5 to 5 in steps of 0.25
4. pntjup = 999 not used.
5. pntjdn = 999 not used.
6. djp = 1 not used
7. logofcls = 1
8. nsamp = 512
9. Xn = 1
10. myStartTime = 830
11. Xopn = 0
12. Xtime = 1420
13. XonCls = 0
14. xmult = 1
15. prc = c

This will produce 1805 different cases or combinations of the input parameters for each of the 104 PWFO output files.

Walk Forward Out-of-Sample Testing

Walk forward analysis attempts to minimize the curve 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 test and out-of-sample sections. Usually for any system, one has some performance metric selection procedure, which we will call a *filter*, used to select the input parameters from the 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 test section output and its associated strategy input parameters. Now suppose we ran our optimization on each of our many test sections and applied our filter to each test section output. We would then use the strategy input parameters found by the *filter* in each test section on the out-of-sample section immediately following that test section. The input parameters found in each test section and applied to each out-of-sample section would produce independent net profits or losses for each of the out-of-sample sections. Using this method we now have "x" number of independent out-of-sample section profit 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 our system will perform on average. Due to the Central Limit Theorem, as our sample size increases, the spurious noise results in the out-of-sample section performance tend to average out to zero in the limit leaving us with what to expect from our system 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 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*"? The 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, for example, 5000 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, 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 system optimization or combinatorial searches with no out-of-sample testing cause losses when traded in real time from something that looked great in the test section. Unfortunately it is human nature to extrapolate past performance to project future trading results and thus 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 in the test section on out-of-sample data. In addition, we must perform the test/out-of-sample analysis many times. Why not just do the out-of-sample analysis once? Well just as in Poker or any card game, where there is considerable variation in luck from hand to hand, walk forward out-of-sample analysis give considerable variation in week-to-week out-of-sample profit “luck”. That is, by pure chance we may have chosen some input parameter set that did well in the 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 test/oos sections and take an average of our weekly results over all out-of-sample sections. This average gives us an expected weekly return and a standard deviation of weekly returns which allows us to statistically estimate the expected equity and its range for N weeks in the future.

Finding The Strategy Input Parameters in The Walk Forward Test Sections

The PWFO generates a number of performance metrics in the test section. The question we are attempting to answer statistically, is which performance metric or combination of performance metrics (which we will call a *filter*) in the test section will produce strategy inputs that produce statistically valid profits in the out-of-sample section. In other words we wish to find a metric *filter* that we can apply to the test section that can give us strategy inputs that will produce, on average, good trading results in the future. The PWFO produces a total of 32 different performance metrics in the test section. If we have 1805 different input variations or cases then the test section consists of 32 columns of performance metrics for each of the 1805 input cases or rows.

An example of a simple filter would be to choose the row in the test section that had the highest net profit or perhaps a row that had one the best performance metric from one of the other 32 PWFO metrics. 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 test section be greater than 10 trades a month and less than 50. We require this so we can eliminate inputs that produce very few trades and inputs that generate more than 2.5 trades on average a day. One calendar month is approximately 21 trading days so we are requiring at least a trade every other day on average. In addition we are trying to catch the intraday trend and not get whipsawed. Setting the number of trades per month to be less than 50 only choose those inputs which have the tendency to catch the main intraday trend. Not many traders can stay with a strategy that has a large number of losers in a row (LR). For this filter we will choose $LR \leq 7$. This choice of LR is completely arbitrary and is what I feel comfortable with. After using a NT and LR filter, as described, there can still be 100's of rows left in the PWFO file test section. The PWFO generates the metric **dev**. This metric calculates the median of the absolute deviations of equity from straight line fit to equity curve. Thus we would want the median to be as small as possible. We use the median for this metric, because we do not want the statistic distorted by a few outlier trades. The smaller **dev** is, the more efficient the strategy is in generating a smoother equity curve. Let us choose the 10 rows that contain the **smallest(bottom) dev** values from the rows that are left from the NT-LR screen. In other words we sort **dev** from low to high, eliminate the rows that have $NT < 10$ and $NT > 50$ and $LR > 7$ then choose the Bottom 10 Rows of whatever is left. This particular filter will now leave 10 cases or rows in the PWFO file that satisfy the above filter conditions. We call this filter **b10dev|7:10::50** where **b10dev** means the Bottom 10 **dev** rows left *after* the NT-LR filter. Suppose for this filter, within the 10 PWFO rows that are left, we want the row that has the maximum PWFO **mp-rd** metric in the test section. This metric measures the difference between the final profit of each trade and the maximum trade loss (rundown) of the trade. The farther the final trade profit is from the maximum trade drawdown, the better the performance of the input variable. Thus, we would want the median of this metric to be as large as possible.. This would produce a filter named **b10dev|7:10::50-mp-rd**. This filter leaves only one row in the PWFO test section with its associated strategy inputs

and out-of-sample net profit in the out-of-sample section. This particular **b10dev|7:10::50-mp-rd** filter is then calculated for each of the PWFO files and the average out-of-sample performance is calculated. 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 each filter's total out-of-sample *net* profits by chance. By *net* we mean subtracting the cost and slippage of all round trip trades from the total out-of-sample profits. Here is how the bootstrap technique is applied. Suppose as an example, we calculate the total out-of-sample net profits(tOnpNet) of 5000 different **TopN-Metric- LR-NT** filters. A mirror filter is created for each of the 5000 filters. However, instead of picking an out-of-sample net profit(OSNP) from a filter row, 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 PWFO 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 tOnPNets will allow us to calculate the chance probability of each **TopN-Metric- LR-NT** filter's tOnpNet. Thus given the mirror filter's bootstrap random tOnpNet average and standard deviation, we can calculate the probability of obtaining the TopN-Metric-PF-LR-NT filter's tOnpNet by pure chance alone. Since for this run there are 3844 different filters, we can calculate the expected number of cases that we could obtain by pure chance that would match or exceed the tOnpNet of the filter we have chosen or $(3844) \times (\text{tOnpNet Probability})$. 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 \$59260 is $3844 \times 6.13 \times 10^{-5} = 0.236$. This is much less than one case so it is improbable that our result was due to pure chance.

The partial run shown in Figure 3 reveals that the following filter will produce the most consistent and reliable out-of-sample results.

Filter: #Trds>=10 and #Trds<=50 and LR<=7 and Bottom 10 dev then maximum mp-rd

Where:

- **NT** = The number of trades for a given set of inputs in the test section.
- **LR** = Maximum losing trades in a row for a given set of inputs in the test section
- **dev** = Median of the absolute deviations of equity from a straight line fit to equity curve..
- **mp-rd** = Median of all Trades(Trade Profit - Maximum Trade Rundown) for a given set of inputs in the test section

The first part of the filter chooses those rows or cases out of the 1805 rows in each PWFO file test(in-sample) section that satisfy the criteria **NT<=50 and NT>=10**. After using a NT and LR filter, there can still be 100's of rows left in the PWFO file. The PWFO generates the metric **dev**, which is the Median of the absolute deviations of equity from straight line fit to equity curve for a particular set of inputs in the test section. Let us choose the 10 rows that contain the **Bottom 10 dev** values from the rows that are left from the **NT-LR** screen.. This particular filter will now leave 10 cases or rows in the PWFO file that satisfy these filter conditions. We call this filter **b10dev|7:10::50** where b10dev means the Bottom 10 dev rows left *after* the NT-LR filter. Within the 10 PWFO rows that are left, we want the row that has the maximum PWFO metric, **mp-rd**, in the test section. This **Filter** or selection procedure will leave only one choice for the system input values of *thres, pntup, pntdn*. We then use these input values found in the test section by the **Filter** on the **next week** of five minute CL **out-of-sample** price bars **following** the test section.

Results

Table 1 below presents a table of the 104 test and out-of-sample windows, the selected optimum parameters and the weekly out-of-sample profit/loss results using the filter described above.

Figure 1 presents a graph of the equity curve generated by using the filter on the 104 weeks of 6/27/08 – 6/18/10. The equity curve is plotted from NetEq in Table 1. Plotted on the equity curve is the least squares straight line. The

weekly CL prices are also included on the plot so that we can compare how the equity curve changes vs CL price changes.

Figure 2 presents the out-of-sample 5 minute bar chart of CL for 6/16/10 to 6/18/10 with the EPFFT Indicator and all the buy and sell signals for those dates.

Figure 3 Partial output of the Walk Forward Metric Performance Explorer

Discussion of System Performance

In Figure 3 Row 2 of the spreadsheet filter output are some statistics that are of interest for our filter. An interesting statistic is **Blw**. Blw is the maximum number of weeks the OSNP equity curve failed to make a new high. Blw is 18 weeks for this filter. This means that 18 weeks was the longest time that the equity for this strategy failed to make a new equity high.

To see the effect of walk forward analysis, take a look at **Table 1**. Notice how the input parameters *thres*, *pntup* and *pntdn* take sudden jumps from high to low and back. This is the walk forward process quickly adapting to changing volatility conditions in the test sample. In addition, notice how often *thres* changes from 4 to 20. When the data gets very noisy with a lot of spurious price movements, the threshold to include frequencies has to be higher. During other times when the noise level is not as much the threshold to include frequencies can be lower to get the higher FFT frequencies and get onboard a trend faster. This is what the threshold filter is doing.

In Figure 1, which presents a graph of the equity curve using the filter on the 104 weeks of 6/27/08 – 6/18/10, notice how the equity curve follows the trend line with an R^2 of 0.81. The weekly plot of CL prices is superimposed on the equity curve plot. Notice how fast the equity curve is able to adapt with changing trends in CL. From 6/27/08 to 3/1/09 CL fell fast from 180 to 60. from 3/1/09 to 6/18/10 CL slowly climbed from 60 to 90. Yet despite these changing price trends the EPFFT strategy was able to adapt and follow the trend line.

Using this filter, the strategy was able to generate \$59,260 net equity after commissions and slippage trading one CL contract for 104 weeks. This period of time from 6/27/08 to 6/18/10 was one of the most volatile markets in history. Yet the EPFFT strategy was able to adapt quite well. Note \$30 roundtrip commission and slippage was subtracted from each trade and no positions were carried over night. From Table 1, the largest net losing week was -\$10480 on 10/10/08 a very wild financial time and market week. The largest drawdown was -\$22480 during that same financial crises in the U.S. However this only lasted four weeks from 10/3/08 to 10/24/08 and completely recovered and made a new equity high in five weeks. The longest time between new equity highs was 18 weeks.

In observing Table 1 we can see that this strategy and filter made trades from a low of 6 trades/week to a high of 17 trades/week with an average of 9.7 trades/week.

Given 24 hour trading of the Crude Light, restricting the strategy to trade only from 830am to 2:20pm ET caused the strategy to miss many profitable trends opportunities when Asia and then Europe opened trading in the early morning. Further research will include the A.M. time zones.

References

1. Efron, B., Tibshirani, R.J., (1993), "An Introduction to the Bootstrap", New York, Chapman & Hall/CRC.
2. Morrison, Norman "Introduction to Sequential Smoothing and Prediction", McGraw-Hill Book Company, New York, 1969.

Table 1 Walk Forward Out-Of-Sample Performance Summary for CL End Point Fast Fourier Transform System

CL-5 min bars 5/21/2008 - 6/18/2010. The input values *thres*, *pntup*, *pntdn* are the values found from applying the filter to the test section optimization runs.

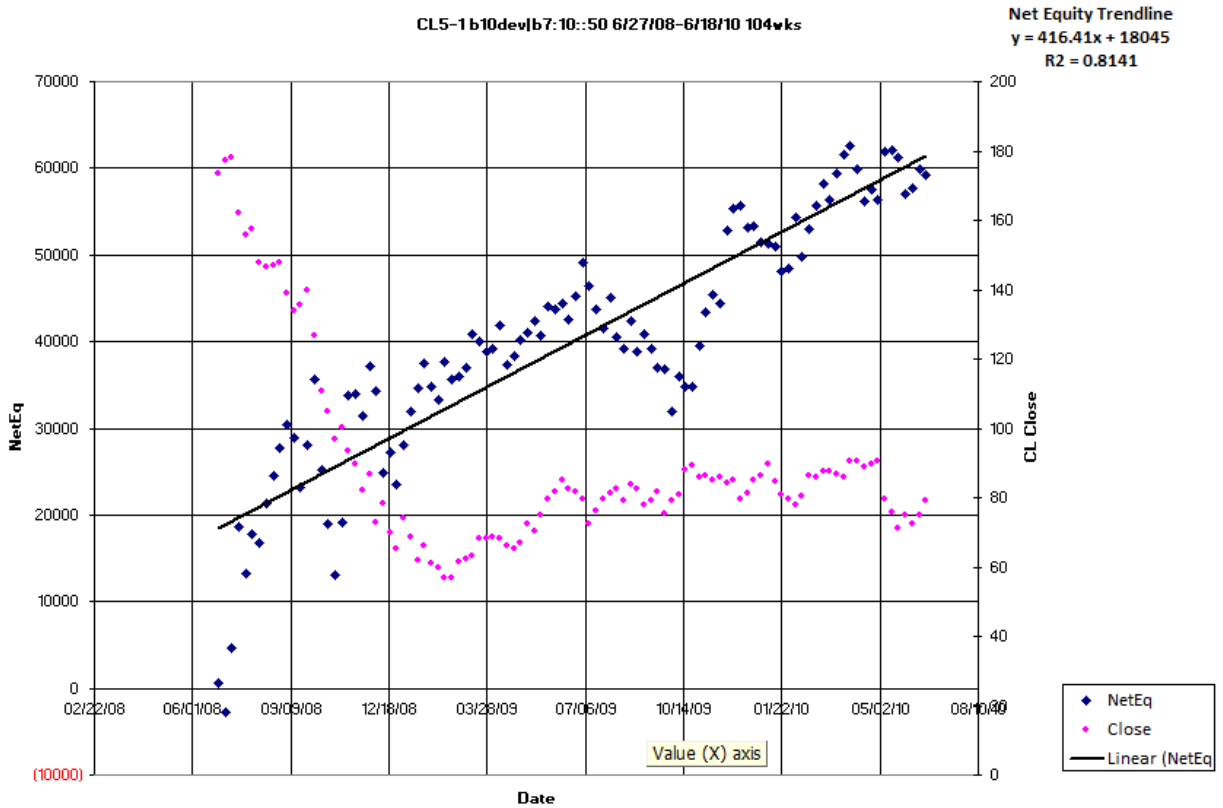
Filter= NT<=50 and NT>=10, LR<=7 and Bottom 10 dev, maximum mp-rd

Test Dates		Out-Of-Sample Dates		osnp	NOnp\$30	NetEq	oilt	odd	ont	avosnp	thres	pntup	pntdn	CL Close
05/21/08	to 06/20/08	06/23/08	to 06/27/08	1040	680	680	-1490	-3590	12	86.7	8	4	2.25	173.35
05/28/08	to 06/27/08	06/30/08	to 07/04/08	(3130)	(3370)	(2690)	-1790	-4700	8	-391	8	5	2.25	177.32
06/04/08	to 07/04/08	07/07/08	to 07/11/08	7530	7350	4660	-580	-580	6	1255	20	5	0.75	178.22
06/11/08	to 07/11/08	07/14/08	to 07/18/08	14110	13900	18560	-780	-780	7	2016	16	4.75	1	162.02
06/18/08	to 07/18/08	07/21/08	to 07/25/08	(5170)	(5380)	13180	-2890	-5170	7	-739	20	3.5	5	155.81
06/25/08	to 07/25/08	07/28/08	to 08/01/08	4930	4690	17870	-2300	-3050	8	616.2	12	4.25	2.25	157.65
07/02/08	to 08/01/08	08/04/08	to 08/08/08	(670)	(1060)	16810	-1460	-3180	13	-51.5	8	4.25	2	147.75
07/09/08	to 08/08/08	08/11/08	to 08/15/08	4730	4490	21300	-920	-920	8	591.2	16	3.75	1	146.32
07/16/08	to 08/15/08	08/18/08	to 08/22/08	3460	3220	24520	-2820	-4320	8	432.5	12	4.5	1.25	147.12
07/23/08	to 08/22/08	08/25/08	to 08/29/08	3570	3270	27790	-1340	-2190	10	357	20	4.75	1.25	147.99
07/30/08	to 08/29/08	09/01/08	to 09/05/08	3000	2700	30490	-1590	-2590	10	300	20	2.5	0.5	138.76
08/06/08	to 09/05/08	09/08/08	to 09/12/08	(1190)	(1580)	28910	-1650	-4170	13	-91.5	20	2.25	1	133.71
08/13/08	to 09/12/08	09/15/08	to 09/19/08	(5480)	(5810)	23100	-1850	-7260	11	-498	20	4.5	0.5	135.62
08/20/08	to 09/19/08	09/22/08	to 09/26/08	5370	5040	28140	-910	-1470	11	488.2	16	1.25	3.75	139.76
08/27/08	to 09/26/08	09/29/08	to 10/03/08	7650	7470	35610	-1290	-2460	6	1275	20	4.25	2.25	126.75
09/03/08	to 10/03/08	10/06/08	to 10/10/08	(10120)	(10480)	25130	-3380	-10920	12	-843	20	5	4.25	110.57
09/10/08	to 10/10/08	10/13/08	to 10/17/08	(5800)	(6160)	18970	-3500	-7710	12	-483	20	1.5	3	104.72
09/17/08	to 10/17/08	10/20/08	to 10/24/08	(5540)	(5840)	13130	-2250	-6210	10	-554	16	4.5	3.5	96.74
09/24/08	to 10/24/08	10/27/08	to 10/31/08	6330	6060	19190	-1240	-1240	9	703.3	20	4.5	2.5	100.4
10/01/08	to 10/31/08	11/03/08	to 11/07/08	14810	14570	33760	-570	-1060	8	1851	16	3.75	3.75	93.63
10/08/08	to 11/07/08	11/10/08	to 11/14/08	360	120	33880	-1790	-2000	8	45	20	4.25	5	89.63
10/15/08	to 11/14/08	11/17/08	to 11/21/08	(2140)	(2530)	31350	-1370	-3950	13	-165	16	0.75	5	82.15
10/22/08	to 11/21/08	11/24/08	to 11/28/08	6140	5810	37160	-1910	-1910	11	558.2	16	1.25	4.5	86.65
10/29/08	to 11/28/08	12/01/08	to 12/05/08	(2570)	(2930)	34230	-1980	-4580	12	-214	20	2.75	4	73.03
11/05/08	to 12/05/08	12/08/08	to 12/12/08	(8840)	(9410)	24820	-2760	-8840	19	-465	12	5	3.75	78.5
11/12/08	to 12/12/08	12/15/08	to 12/19/08	2800	2470	27290	-1140	-1510	11	254.5	20	4.25	1.25	70.03
11/19/08	to 12/19/08	12/22/08	to 12/26/08	(3580)	(3850)	23440	-1720	-5350	9	-398	20	5	1.5	65.38
11/26/08	to 12/26/08	12/29/08	to 01/02/09	4830	4590	28030	-1780	-1780	8	603.8	16	5	5	74.01
12/03/08	to 01/02/09	01/05/09	to 01/09/09	4240	3970	32000	-1590	-1590	9	471.1	16	5	5	68.5
12/10/08	to 01/09/09	01/12/09	to 01/16/09	2910	2700	34700	-1150	-1290	7	415.7	20	4.5	0.75	62.1
12/17/08	to 01/16/09	01/19/09	to 01/23/09	3200	2780	37480	-1460	-1720	14	228.6	16	4	0.75	66
12/24/08	to 01/23/09	01/26/09	to 01/30/09	(2270)	(2690)	34790	-2400	-4780	14	-162	20	4.25	2	61.21
12/31/08	to 01/30/09	02/02/09	to 02/06/09	(1200)	(1470)	33320	-980	-2060	9	-133	20	2.25	4.75	59.7
01/07/09	to 02/06/09	02/09/09	to 02/13/09	4700	4400	37720	-1240	-1240	10	470	20	3.75	3.5	57.04
01/14/09	to 02/13/09	02/16/09	to 02/20/09	(1820)	(2090)	35630	-1920	-3910	9	-202	20	2.5	3.75	56.77
01/21/09	to 02/20/09	02/23/09	to 02/27/09	540	270	35900	-1590	-3260	9	60	16	5	5	61.5
01/28/09	to 02/27/09	03/02/09	to 03/06/09	1240	1060	36960	-1100	-1120	6	206.7	20	4.5	3.25	62.26
02/04/09	to 03/06/09	03/09/09	to 03/13/09	4180	3880	40840	-2200	-2570	10	418	20	3.75	3.25	62.99
02/11/09	to 03/13/09	03/16/09	to 03/20/09	(510)	(870)	39970	-1540	-5600	12	-42.5	20	1.25	5	68.05
02/18/09	to 03/20/09	03/23/09	to 03/27/09	(830)	(1190)	38780	-760	-2530	12	-69.2	20	0.75	5	68.36
02/25/09	to 03/27/09	03/30/09	to 04/03/09	820	460	39240	-1170	-1370	12	68.3	20	2.5	3	68.49
03/04/09	to 04/03/09	04/06/09	to 04/10/09	2800	2560	41800	-360	-630	8	350	16	4	1	68.22
03/11/09	to 04/10/09	04/13/09	to 04/17/09	(4030)	(4450)	37350	-1820	-4500	14	-288	12	2	4	66.31
03/18/09	to 04/17/09	04/20/09	to 04/24/09	1220	980	38330	-340	-660	8	152.5	12	5	0.75	65.39
03/25/09	to 04/24/09	04/27/09	to 05/01/09	2160	1920	40250	-560	-680	8	270	20	0.5	5	67.04
04/01/09	to 05/01/09	05/04/09	to 05/08/09	1030	790	41040	-960	-1250	8	128.8	12	4.75	3.25	72.47
04/08/09	to 05/08/09	05/11/09	to 05/15/09	1530	1290	42330	-1070	-1810	8	191.2	20	4.75	4.25	70.18

04/15/09	to	05/15/09	05/18/09	to	05/22/09	(1330)	(1630)	40700	-970	-2670	10	-133	12	3.75	3.5	74.85
04/22/09	to	05/22/09	05/25/09	to	05/29/09	3470	3290	43990	-120	-160	6	578.3	20	1.5	4.75	79.49
04/29/09	to	05/29/09	06/01/09	to	06/05/09	120	(180)	43810	-1390	-2980	10	12	16	0.5	4.5	81.62
05/06/09	to	06/05/09	06/08/09	to	06/12/09	880	640	44450	-800	-1510	8	110	16	0.5	4.5	85.22
05/13/09	to	06/12/09	06/15/09	to	06/19/09	(1560)	(1920)	42530	-1330	-2570	12	-130	12	1.25	4.25	82.66
05/20/09	to	06/19/09	06/22/09	to	06/26/09	2880	2670	45200	-500	-1330	7	411.4	12	3.75	3.5	81.8
05/27/09	to	06/26/09	06/29/09	to	07/03/09	4090	3880	49080	-1070	-1540	7	584.3	12	1.25	3.5	79.37
06/03/09	to	07/03/09	07/06/09	to	07/10/09	(2180)	(2690)	46390	-990	-4160	17	-128	8	2	3.25	72.53
06/10/09	to	07/10/09	07/13/09	to	07/17/09	(2410)	(2740)	43650	-1160	-2410	11	-219	16	4.25	1.75	76.2
06/17/09	to	07/17/09	07/20/09	to	07/24/09	(1830)	(2130)	41520	-1000	-2910	10	-183	20	2	2.5	79.67
06/24/09	to	07/24/09	07/27/09	to	07/31/09	3770	3560	45080	-830	-830	7	538.6	16	2.5	2.75	81.07
07/01/09	to	07/31/09	08/03/09	to	08/07/09	(4220)	(4610)	40470	-1370	-4610	13	-325	12	2	3.25	82.55
07/08/09	to	08/07/09	08/10/09	to	08/14/09	(880)	(1210)	39260	-1180	-3190	11	-80	12	1.25	4.5	79.13
07/15/09	to	08/14/09	08/17/09	to	08/21/09	3400	3160	42420	-780	-1510	8	425	4	2.5	5	83.61
07/22/09	to	08/21/09	08/24/09	to	08/28/09	(3190)	(3610)	38810	-1300	-4470	14	-228	4	2.5	5	82.46
07/29/09	to	08/28/09	08/31/09	to	09/04/09	2360	2090	40900	-550	-670	9	262.2	20	3.75	1	77.74
08/05/09	to	09/04/09	09/07/09	to	09/11/09	(1280)	(1700)	39200	-1710	-4520	14	-91.4	20	2.75	1	79.01
08/12/09	to	09/11/09	09/14/09	to	09/18/09	(1960)	(2200)	37000	-1210	-2230	8	-245	16	4.75	0.5	81.76
08/19/09	to	09/18/09	09/21/09	to	09/25/09	120	(150)	36850	-1700	-1740	9	13.3	16	3.5	5	75.29
08/26/09	to	09/25/09	09/28/09	to	10/02/09	(4540)	(4930)	31920	-1810	-5060	13	-349	12	3.75	4.5	79.22
09/02/09	to	10/02/09	10/05/09	to	10/09/09	4400	4100	36020	-760	-920	10	440	16	1.25	4.75	81.04
09/09/09	to	10/09/09	10/12/09	to	10/16/09	(930)	(1230)	34790	-980	-2670	10	-93	4	3	4.75	87.8
09/16/09	to	10/16/09	10/19/09	to	10/23/09	270	30	34820	-1630	-2390	8	33.8	12	4.5	0.5	89.28
09/23/09	to	10/23/09	10/26/09	to	10/30/09	5050	4780	39600	-1020	-2220	9	561.1	12	3.75	2.25	85.78
09/30/09	to	10/30/09	11/02/09	to	11/06/09	4030	3850	43450	-620	-620	6	671.7	20	2.5	3.25	86.21
10/07/09	to	11/06/09	11/09/09	to	11/13/09	2220	1980	45430	-820	-820	8	277.5	8	4.25	3.5	85.13
10/14/09	to	11/13/09	11/16/09	to	11/20/09	(670)	(970)	44460	-1150	-1460	10	-67	8	3.75	4	85.73
10/21/09	to	11/20/09	11/23/09	to	11/27/09	8530	8320	52780	-310	-310	7	1219	12	3.5	3.5	84.31
10/28/09	to	11/27/09	11/30/09	to	12/04/09	2910	2640	55420	-1540	-2270	9	323.3	16	1.5	4.25	85.16
11/04/09	to	12/04/09	12/07/09	to	12/11/09	640	340	55760	-700	-1460	10	64	8	3	2.75	79.56
11/11/09	to	12/11/09	12/14/09	to	12/18/09	(2190)	(2520)	53240	-1350	-3080	11	-199	8	3	2.75	81.25
11/18/09	to	12/18/09	12/21/09	to	12/25/09	290	80	53320	-1280	-1390	7	41.4	12	4	1.75	84.88
11/25/09	to	12/25/09	12/28/09	to	01/01/10	(1540)	(1840)	51480	-840	-2220	10	-154	12	4	0.5	86.19
12/02/09	to	01/01/10	01/04/10	to	01/08/10	90	(180)	51300	-860	-900	9	10	12	2.5	3.25	89.58
12/09/09	to	01/08/10	01/11/10	to	01/15/10	(140)	(350)	50950	-710	-1850	7	-20	12	4.5	2.5	84.83
12/16/09	to	01/15/10	01/18/10	to	01/22/10	(2530)	(2800)	48150	-1460	-4540	9	-281	12	4.75	2.5	81.04
12/23/09	to	01/22/10	01/25/10	to	01/29/10	550	310	48460	-610	-1000	8	68.8	16	3.5	3	79.39
12/30/09	to	01/29/10	02/01/10	to	02/05/10	6040	5800	54260	-1020	-1850	8	755	4	4.75	3.25	77.69
01/06/10	to	02/05/10	02/08/10	to	02/12/10	(4000)	(4450)	49810	-1530	-4000	15	-267	16	2.5	0.5	80.63
01/13/10	to	02/12/10	02/15/10	to	02/19/10	3310	3100	52910	-890	-890	7	472.9	16	4	5	86.2
01/20/10	to	02/19/10	02/22/10	to	02/26/10	2860	2710	55620	-180	-180	5	572	16	4	3.5	85.8
01/27/10	to	02/26/10	03/01/10	to	03/05/10	2720	2540	58160	-520	-520	6	453.3	16	3.5	3.75	87.64
02/03/10	to	03/05/10	03/08/10	to	03/12/10	(1520)	(1820)	56340	-730	-2460	10	-152	4	3.25	4.75	87.38
02/10/10	to	03/12/10	03/15/10	to	03/19/10	3190	3040	59380	-960	-960	5	638	12	4.25	1.75	86.77
02/17/10	to	03/19/10	03/22/10	to	03/26/10	2520	2280	61660	-400	-560	8	315	12	0.5	4.5	85.8
02/24/10	to	03/26/10	03/29/10	to	04/02/10	1120	910	62570	-460	-1170	7	160	16	1	3.75	90.67
03/03/10	to	04/02/10	04/05/10	to	04/09/10	(2360)	(2690)	59880	-890	-3380	11	-215	8	2	3.5	90.72
03/10/10	to	04/09/10	04/12/10	to	04/16/10	(3240)	(3630)	56250	-1550	-3550	13	-249	16	0.75	3.75	89.04
03/17/10	to	04/16/10	04/19/10	to	04/23/10	1580	1340	57590	-870	-1780	8	197.5	4	3.25	4.5	89.49
03/24/10	to	04/23/10	04/26/10	to	04/30/10	(830)	(1160)	56430	-770	-2260	11	-75.5	8	1.5	3.5	90.52
03/31/10	to	04/30/10	05/03/10	to	05/07/10	6000	5550	61980	-740	-1170	15	400	8	3.75	1	79.48
04/07/10	to	05/07/10	05/10/10	to	05/14/10	460	130	62110	-1300	-2210	11	41.8	16	4.75	1	75.98
04/14/10	to	05/14/10	05/17/10	to	05/21/10	(670)	(940)	61170	-1530	-2760	9	-74.4	20	4.75	1.75	71.12
04/21/10	to	05/21/10	05/24/10	to	05/28/10	(3810)	(4170)	57000	-1130	-5230	12	-318	8	5	1.75	75.05
04/28/10	to	05/28/10	05/31/10	to	06/04/10	1070	740	57740	-1580	-4060	11	97.3	20	1.25	2.5	72.59
05/05/10	to	06/04/10	06/07/10	to	06/11/10	2330	2090	59830	-620	-1090	8	291.2	20	2.75	2	74.86
05/12/10	to	06/11/10	06/14/10	to	06/18/10	(360)	(570)	59260	-1320	-1660	7	-51.4	20	1.75	3.75	79.34

osnp = Weekly Out-of-sample gross profit in \$
NOnp\$30 = Weekly Out-Of-Sample Net Profit in \$ = **osnp-ont*30**.
NetEq = running sum of the weekly out-of-sample net profits in \$
ollt = The largest losing trade in the out-of-sample section in \$.
odd = The close drawdown in the out-of-sample section in \$.
ont = The number of trades in the out-of-sample week.
avosnp = The average out-of-sample profit per trade for that week \$

**Figure 1 Graph of Net Equity Applying the Walk Forward Filter Each Week
On CL 5min Bar Prices 6/27/08 – 6/18/10**



**Figure 2 Walk Forward Out-Of-Sample Performance Summary for CL EPFFT System 5 minute bar chart
from 6/16/10-6/18/2010**

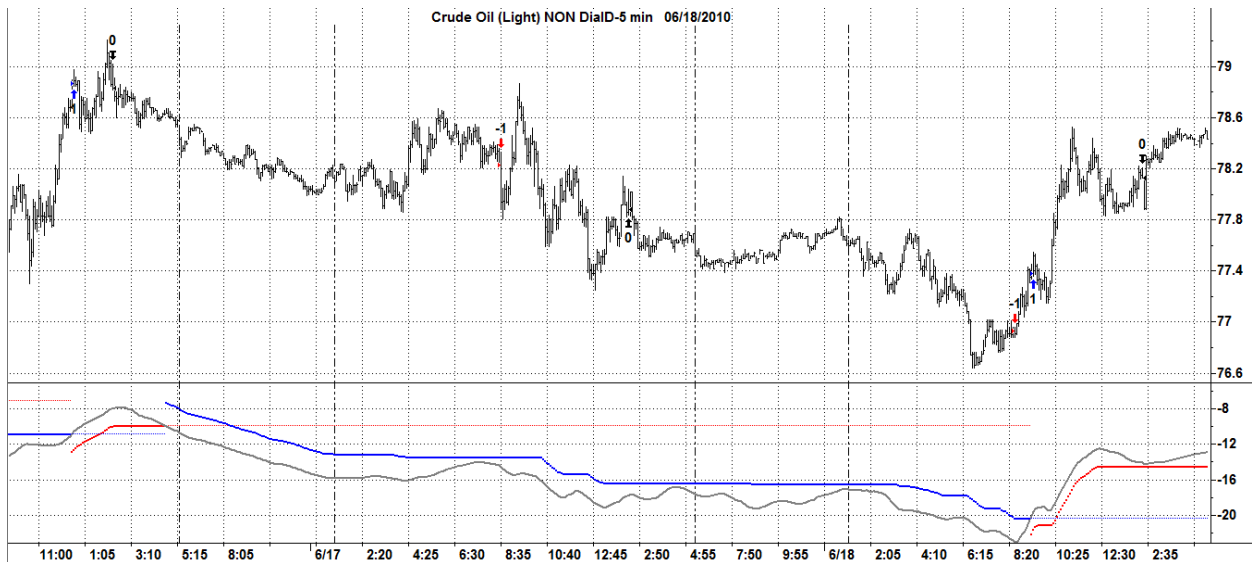


Figure 3 Partial output of the Walk Forward Metric Performance Explorer (WFMPE)

1	CL5-1epfft	d080627	d100618	AnyTnp	nT<50	t104					a(45844)	s27364	f3844							c=530	
2	Filter	Metric	tOnp	mOnp	aOnp	aanp	aOnT	B0	%P	t	std	llw	eqDD	lr	#	b00	Blw	BE	ndd	tOnpNet	Prob
3	b10dev 7:10::50	mp-rd	89440	595	860	88.9	9.7	(11.3)	0.59	2.21	3963.6	-10120	-21460	3	104	969	18	85	36.2	59260	6.13E-05
4	b10dev 7::50	mp-rd	89440	595	860	88.9	9.7	(11.3)	0.59	2.21	3963.6	-10120	-21460	3	104	969	18	85	36.2	59260	6.13E-05
5	b10dev a::50	mp-rd	86850	590	835	86.0	9.7	(11.4)	0.58	2.14	3973.9	-10120	-21460	3	104	939	18	90.6	37.9	56550	9.13E-05
6	b10dev a:10::50	mp-rd	86850	590	835	86.0	9.7	(11.4)	0.58	2.14	3973.9	-10120	-21460	3	104	939	18	90.6	37.9	56550	9.13E-05

The WFMPE Filter Output Columns are defined as follows

Row 1 CL5-1epfft is the PWFO Stub, File Start Date(6/27/08), File End Date(6/18/10), Number of weeks(t104) a=average of bootstrap random picks. S= standard deviation of bootstrap random picks. F=number of different filters examined. C= slippage and round trip trade cost(c=\$30).

Filter = The filter that was run. For example, b10dev|7:10::50 - mp-rd

Metric = The PWFO performance . For this filter, b10dev|7:10::50
Metric=mp-rd (median of final trade profit/loss-maximum rundown of trade).

This b10dev|7:10::50 -mp-rd filter produced the following average 104 week statistics on this line(row 3).

tOnp = Total out-of-sample(oos) net profit for these 104 weeks.

mOsp = median oos net profit for the 104 weeks

aOsp = Average oos net profit for the 104 weeks

aanp = Average oos profit per trade

aOnT = Average number of oos trades per week

B0 = The 104 week trend of the out-of-sample weekly profits

%P = The percentage of oos weeks that were profitable

t = The student t statistic for the 104 weekly 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 104 weekly oos profits

llw = The largest losing oos week

eqDD = The oos equity drawdown

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

= The number of weeks this filter produced a weekly result. Note for some weeks 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 \$/week.

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

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

ndd = The normalized equity drawdown = 100*eqDD/tONet

tONet = Total out-of-sample net profit(tOnp) minus the total trade cost. tONet=tOnp - #*aOnT*Cost.

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