Trading The Russell 2000 5min Bars Using The Adaptive Goertzel DFT System

Working Paper September 2012

In a previous working paper entitled "MESA vs Goertzel DFT", http://www.meyersanalytics.com/publications/MesaVsGDFT.pdf we demonstrated that the Goertzel Algorithm, a subset of the Discrete Fourier Transform (DFT), has better frequency detection abilities of sine waves imbedded in noise than the MESA algorithm when the noise amplitude is equal to or greater than the signal amplitude. MESA, which stands for Maximum Entropy Spectral Analysis, is more commonly referred to as the Burg AR algorithm in engineering and is a widely used mathematical technique designed to find the frequencies in data. MESA was developed by J.P Burg for his Ph.D dissertation at Stanford University in 1975. The use of the MESA technique for stocks and futures has been written about in many articles.

Previous researchers using MESA constrained themselves to using MESA to find only the cycle with the highest amplitude and called that cycle the dominant cycle. In this paper we will find and use the frequencies with the five or ten highest amplitudes to create a noise filtered signal curve that we will follow to create our strategy buy and sell signals.

The Goertzel Algorithm.

The value of the Discrete Fourier Transform is that for N input data points, the DFT can not only find the frequencies in the data but also the amplitude and phases of the sine waves in the data at the discrete frequency points of 1/N, 2/N,…. to (N/2)/N. For instance if N, the number of price bars or closes, were equal to 20 we could find the amplitude and phases for the 20 period cycle (20 bars/ cycle), the 10 period cycle, the 6 2/3 period cycle all the way down to the 2 period cycle. We could then find the ten cycles with the highest amplitudes and reconstruct a new signal with those ten periods, amplitudes and phases. Remember, cycle= 1/period. As traders we are more accustomed to thinking of terms of periods like the 3 day cycle (frequency = 1/3 cycle/day) or the 10 bar cycle (frequency = 1/10 cycles/bar) etc. Unfortunately the DFT can only calculate equally spaced frequencies of 1/N. In the above example, where N=20, using the DFT we could only calculate the amplitudes of the periods (periods=1/frequency) of 20, 10, 6 2/3 etc. What if the true signal period was between 20 and 10? The DFT couldn't find it. That was the advantage of MESA. MESA was not constrained to the 1/N spacing. With MESA any grid of frequencies could be examined. Fortunately, as shown in my previous article, a special subset of the discrete Fourier transform called the Goertzel algorithm can be used to find frequencies in between the 1/N frequency divisions. The Goertzel algorithm is used extensively in telephone tone detection. We all are familiar with cellular phones. When you press a button have you ever wondered how the telephone company knows what button you pushed? The answer is the Goertzel algorithm. This algorithm is built into tiny integrated circuits and immediately (within milliseconds) detects the tone of the button you pushed. Here we will use the Goertzel algorithm to detect the frequencies of the price series we give it.

Despite the advantage of the Goertzel algorithm frequency detection abilities, it has three drawbacks. One drawback of the Goertzel algorithm is that it is much slower than the Fast Fourier transform. If we had 512 data points and we wanted to look at 128 different frequencies, the FFT computation would be proportional to $512*\log_2(512)= 4608$ operations while the Goertzel computation would be proportional $512*128 = 65536$ operations. In other words, in this case, the Goertzel algorithm would take more than 10 times longer than the FFT to compute. However, as a computation time comparison, Goertzel would take about half the time as MESA to compute those 128 frequencies. The second drawback of the Goertzel algorithm is that in order to find a frequency in Goertzel when the noise amplitude is high you need enough data so that your lowest frequency (largest period) is able to complete at least 3 cycles. This means that if we were examining daily closing prices and we wanted to find periods of 75 days and less than we would need at least 3*75=225 days of prices in order to detect that period in noisy data. The third drawback of the Goertzel algorithm is that while it can detect the frequency within the 1/N spacing it cannot detect more than one frequency within that spacing. For instance if N=20, and we are looking for a frequency between 1/20 and 2/20, the Goertzel can detect the frequency anywhere at or between these two frequencies. However, if there were a

second frequency in-between these two frequencies, Goertzel could not find it. If there were two frequencies between these two values, Goertzel would produce one frequency as a weighted average of the two. This is where MESA has a clear advantage. On data with a low noise component, Mesa could detect these two closely spaced frequencies. However, if the data is very noisy than MESA would not be able to detect the closely spaced frequencies either.

The Adaptive n Cycle Goertzel-DFT System.
The nature of intraday price movements is constantly changing due to current economic surprises, events and trader sentiment. Also the time of year changes the nature of intraday markets, such as the seasons, holidays, vacation time, etc. As such, the periods or frequencies found on intraday prices 3 weeks ago may no longer be the same as the frequencies found on today's intraday data. We would expect the frequencies found on intraday data to vary over time.

For this system we will create an indicator that walks forward one price bar at a time. The indicator will take a fixed number, N, of closing prices and use the Goertzel Algorithm (GA) to find the M frequencies with the highest amplitudes. Using those frequencies's amplitudes and phases, we will construct a new price that forecasts the price one bar ahead. We will save this next bar forecast value, or forecast point. Next we will move the N closing prices forward one bar adding the next bar and dropping the last bar so that we have exactly N closing prices again. With this new N closing prices window we compute the next bar forecast value and save it. We keep marching the N closing price window forward one bar at a time, calculate and save the new forecast point until we reach the end of our data. We will then connect all the generated forecast points to produce a curve that creates the next bar forecast as the m frequencies used to create the next bar forecast change over time. Thus this curve adapts to the closing prices changing frequencies and projects one day ahead so that its lag is minimized when things change.

For example suppose we had daily data from 7/16/12 to 7/23/12. We would calculate and save the next day forecast value of 7/23/12 and slide our data window up one day from 7/17/12 to 7/24/12. We would calculate the next days forecast value of this new data window and save the forecast value of 7/24/12. We would keep sliding our data window forward one day at a time, calculate the next day's forecast value and save that forecast point. When we reached the end of our data, we would then connect all these saved forecast points to create a new curve which we would follow and use to create the system buy and sell signals.

Adaptive Goertzel DFT System Construction Details
Unfortunately constructing the ***n*** cycle DFT of a price data series is not quite as simple as just taking N closing prices, and directly plugging them into a Goertzel algorithm.

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:

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

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

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. Fortunately we won't be looking for frequencies in that range so this distortion will have minimal impact.

n Cycle Goertzel-DFT Curve Construction

Before we start, we have to determine the largest period we will be looking to include in the ***n*** cycle Goertzel DFT construction. For intraday data the 3-day and 1 day cycles are very important. In this paper we will use 5-minute bars of the Russell 2000 Index Futures(TF). The TF trades on the Intercontinental Exchange (ICE) 22 hours a day. For our study we will only look at the US day trading session from 8:30am to 3:15pm CST Monday through Friday. Each day consists of 81 5min bars a day. Thus the 1-day cycle would need 81 bars. In order to detect the 1-day cycle we need the cycle to repeat at least three times. This means we need at least 3*81= 243 data points in order to detect the 1-day cycle using 5-minute bars. The 3-day cycle would need 3*81=243 bars. In order to detect the 3-day cycle using the Goertzel algorithm we need at least 3*243=729 5-min bars. We will use 729 five min bars of TF data which will detect both the 3-day all cycles below 3 days.

We will use the Goertzel algorithm described in our previous article to determine the top ***n*** frequencies with the highest amplitudes. "***n***", the number of frequencies with the highest amplitudes will be one of our optimization parameters. We will also use the Goertzel algorithm to determine the phases of the frequencies with the n highest amplitudes.

Step 1 End flatten the 729 prices using equation (1) above.

Step 2 Use the Goertzel algorithm to calculate the amplitudes for the frequencies of 1/243 down to 1/6. Frequency = 1/period. We are scanning for periods because as traders we think in terms of periods not frequencies. That is, the 1 day cycle, the 20 bar cycle etc. Thus, here we are scanning for frequencies of 243 bars/cycle to 6 bars/cycle. The frequencies of 5 bars/cycle down to 2 bars/cycle move to fast with 5min bars to take advantage trading these cycles. The slippage and commissions would eat up any profits made.

Step 3 Find the ***n*** frequencies with the highest amplitudes, calculate the phases of these frequencies and save these amplitudes (a[i]), phases (phi[i]) and frequencies (f[i]). Where [i] is one of the ***n*** highest amplitudes found.

Step 4 Calculate the forecast next bar value and the end point bar value using the above ***n*** frequencies, amplitudes and phases. The next bar forecast(fp) = $\sum$a[i]*cos(2*PI*f[i]*730 +phi[i]) i=1 to ***n*** and the Endpoint(ep) = $\sum$a[i]*cos(2*PI*f[i]*729 +phi[i]) i=1 to 10 where a[i], f[i] and phi[i] are the *n* frequency, amplitude and phases found.

Step 5 Save the calculated forecast next bar point and the end point values. Call the forecast next point **fp(k)** and the end point **ep(k)** where k denotes the order of the sliding window. That is, the first sliding window k=1, the second, k=2, etc. Slide the 729 bar data window forward one bar, and repeat steps 1 through 4.

Why do we need **fp** and **ep**? 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 forecast next bar point. This random jumping as the data window slides forward in time adds a small random jump to the forecast next bar point curve. Fortunately this jumping can be minimized by creating a curve from the two saved end points, **fp(k)** and **ep(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 **sumv** where

sumv(k) = sumv(k-1) + fp(k)-ep(k)

fp(k)-ep(k) is like a one bar ahead momentum or velocity. This new curve **sumv(k)** is the sum of all the changes in the next bar's ***n*** cycle forecast value fp(k) from the end point n cycle value. This change series minimizes the magnitude jump problem creating a fairly smooth momentum sum curve.

The Adaptive *n* Cycle Goertzel-DFT System Defined

Even though **sumv** 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 TF 5-minute bars.

Buy Rule:

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

Sell Rule:

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

Testing The Adaptive N Cycle Goertzel-DFT (GZ) System Using Walk Forward Optimization

There will be three strategy parameters to determine:

1. ***ncy***, Number of Cycles (for this study we will look at 1(dominant cycle), 3, 5 and 10 cycles).
2. ***pntup***, if **sumv** has moved up by more than the point amount of *pntup* from the lowest low recorded in **sumv** while short then issue a buy signal
3. ***pntdn***, **if sumv** has moved down by more than the point amount *pntdn* from the highest high recorded in **sumv** while long then sell

As mentioned, to test this system we will use five minute bar prices of the mini Russell 2000 index futures contract traded on the Intercontinental Exchange (ICE) and known by the symbol TF for the 105 weeks from August 2, 2010 to August 31, 2012.

We will test this strategy with the TF 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 GZ Strategy on the TF 5min bar data from August 2, 2010 to August 31, 2012.

What Is An In-Sample Section and Out-Of-Sample Section?

Whenever we do a TS combinatorial 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 strategy inputs. If the report is sorted on say the total net profits(***tnp***) performance metric column then the highest ***tnp*** would correspond to a certain set of strategy inputs. This is called an ***In-Sample section***. If we choose a set of strategy inputs from this report based upon some performance metric we have no idea whether these chosen 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 In-Sample section is defined as ***out-of-sample data***. Since the performance metrics generated in the In-Sample 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 In-Sample 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 a single TS optimization run into a number of user selectable In-Sample and out-of-sample sections. The PWFO prints out the In-Sample sample performance **and the out-of-sample performance results**, on one line, for each strategy 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 In-Sample 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 strategy input parameters from the in-sample section 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 In-Sample 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 In-Sample section that have a higher probability of performing well on out-of-sample data (Please see www.meyersanalytics.com/pwfo.php for a listing of these performance metrics).

For our computer run we will have the PWFO breakup the 105 weeks of TF five-minute bar price data into 105 in-sample/out-of sample files. The In-Sample sections will be 30 calendar days and the out-of-sample(oos) section will be the one week following the In-Sample section. The oos week will always end on a Friday as will the 30-day calendar In-Sample section.

The PWFO 105 in-sample/out-of-sample section dates are shown in **Table 1** on page 9 below. We will then use another software product called the Walk Forward Performance Metric Explorer (WFME) on each of the 105 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 ***(ncy, pntup, pntdn)*** 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 combinatorial optimization engine on the 105 weeks of TF 5 min bars with the following ranges for the Goertzel strategy input variables.

1. ncy 1, 3, 5, 10
2. pntup from 1 to 6 steps of 0.25
3. pntdn from 1 to 6 in steps of 25

This will produce 1764 different cases or combinations of the strategy input parameters for each of the 105 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 In-Sample 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 In-Sample section and its associated strategy input parameters. Now suppose we ran our optimization on each of our many In-Sample sections and applied our filter to each In-Sample section output. We would then use the strategy input parameters found by the ***filter*** in each In-Sample section on the out-of-sample section immediately following that in-sample section. The input parameters found in each in-sample 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, profit factor or some other performance metric? Surely the price noise cancels itself out with such a large number of in-sample 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 strategy input parameters on noisy price 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, 2000 different strategy inputs parameter combinations , the best performance parameters will be from those strategy 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 strategy 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 loses when traded in real time from something that looked great in the in-sample 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 in-sample data will produce profits, we must test the strategy input parameters we found in the in-sample section 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? Well just as in Poker or any card game, where there is considerable variation in hand to hand luck, 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 in-sample 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 in-sample/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 out-of-sample equity and its range for N weeks in the future.

Finding The Strategy Input Parameters in The Walk Forward In-Sample Sections

The PWFO generates a number of performance metrics in the in-sample 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 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 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. The PWFO produces a total of 32 different performance metrics in the in-sample section. If we have 1764 different strategy input variations or cases then the in-sample section consists of 32 columns of performance metrics for each of the 1764 input cases or rows.

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 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. A combination of different metric filters can produce good out-of-sample results minimizing spurious price movement biases in the selection of strategy inputs.

Here is a combination ***filter*** that is used in this paper with good out-of-sample success.. The PWFO generates the metric **llt.** This metric calculates the largest losing trade in the in-sample section. The smaller **llt** is, the more efficient the strategy is in generating a smoother equity curve. Let us choose the 20 rows that contain the smallest(bottom) absolute **llt** values. llt is a negative value. If we sort **llt** from high to low, the smallest negative values will be at the top. We then choose the Top 20 Rows . This particular filter will now leave 20 cases or rows in the in-sample section that satisfy the above filter conditions. Suppose for this filter, within the 20 in-sample rows that are left, we want the row that has the maximum PWFO **b1** metric in the in-sample section**.** The **b1 metric is the slope of the 2nd Order polynomial trend line that is fitted to the trade equity plot**. This would produce a filter named **t20llt-b1**. This filter leaves only one row in the PWFO in-sample section with its associated strategy inputs and out-of-sample net profit in the out-of-sample section. This particular **t20llt-b1** filter finds the strategy inputs parameters in each of the 105 in-sample sections and apples these inputs to the out-of-sample section. Using the filter in-sample strategy inputs on the 105 out-of-sample sections, 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 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 an in-sample 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 (tOnp). The average and standard deviation of the 5000 mirror filter's different random tOnps will allow us to calculate the chance probability of our filter's tOnp. Thus given the mirror filter's bootstrap random tOnp average and standard deviation, we can calculate the probability of obtaining our filter's tOnp by pure chance alone. Figure 3 lists the 5000 mirror filter's bootstrap average for our 105 out-of-sample files of $7458 with a bootstrap standard deviation of $19,488 The probability for obtaining our filters net profit of $80,810 is 8.36 $x10^{-5}$ which is 3.76 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 $80,810 is 1922 x 8.36x 10^{-5} = 0.159. where 1922 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.

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

Filter: Top 20 llt(=Bottom 20 absolute values of llt) then maximum b1

Where:

- **llt** = largest losing trade in the in-sample section.
- **b1 =** In-Sample Section Trade Equity 2nd order polynomial trend line coefficient b1.
  Where**:** Equity = b0+b1*t +b2*t^2

The first part of the filter chooses those rows or cases out of the 1764 rows in each PWFO file test(in-sample) . The PWFO generates the metric **llt**, which is the largest losing trade in the in-sample section. Let us choose the 20 rows that contain the **Bottom 20 absolute llt** values. This particular filter will now leave 20 rows in the PWFO file that satisfy this filter condition . Within the 20 PWFO rows that are left, we want the row that has the maximum PWFO metric, **b1,** in the in-sample section. This ***Filter*** or selection procedure will leave only one choice for the in-sample strategy input values of ***ncy, pntup, pntdn***. We then use these input values found in the in-sample section by the **Filter** on the **next week** of TF 5min **out-of-sample** price bars **following** the in-sample section.

Results

Table 1 below presents a table of the 105 in-sample and out-of-sample windows , the selected filter strategy inputs 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 105 weeks of August 2, 2010 to August 31, 2012.. The equity curve is plotted from NetEq in Table 1. Plotted on the equity curve are the least squares straight line and TF weekly values.

Figure 2 presents the out-of-sample 5 minute bar chart of TF for 8/13/12 to 8/31/12 with the GZ 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 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 11 weeks for this filter. This means that 11 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 ***ncy, 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 in-sample sample. In addition, notice how often ***ncy*** changes from 1 to 3 to 5 to 10 and back as short term trend and volatility of TF change in the in-sample section

In Figure 1, which presents a graph of the equity curve using the filter on the 105 out-of-sample weeks of 9/03/10 – 8/30/12, notice how the equity curve follows the trend line with an R^2 of 0.94. The weekly plot of TF prices is superimposed on the equity curve plot. Notice how fast the equity curve is able to adapt with changing trends in TF. From 4/29/11 to 9/23/11 TF fell fast from 845 to 652 and then from 9/23/11 to 10/28/11 TF raced up from 637 to 748. Yet despite these fast moving changing price trends the GZ strategy was able to adapt and follow the trend line.

Using this filter, the strategy was able to generate $80810 out-of-sample net equity trading one CL contract for 105 weeks. The period of time from 4/29/11 to 2/03/12 was a very volatile down and up market. Yet the GZ strategy was able to adapt quite well. . From Table 1, the largest net losing week was -$4320 on the week of 9/12/11 to 9/16/11 a wild financial time and market week. The largest drawdown was -$6710 during whipsaw TF price movements from 10/28/11-11/25/11.

In observing Table 1 we can see that this filter and made trades from a low of 1 trade/week to a high of 16 trades/week with an average of 4.5 trades/week.

Given 24 hour trading of the Mini Russell 2000 index future, restricting the strategy to trade only from 8:30am to 3:15pm CST 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. Burg, J. P., 'Maximum Entropy Spectral Analysis", Ph.D. dissertation, Stanford University, Stanford, CA. May 1975.
2. Goertzel, G., "An Algorithm for the evaluation of finite trigonometric series" American Math Month, Vol 65, 1958 pp34-35.
3. Kay, Steven M., "Modern Spectral Estimation", Prentice Hall, 1988
4. Marple, Lawrence S. Jr., "Digital Spectral Analysis With Applications", Prentice Hall, 1987
5. Meyers, Dennis , "MESA vs Goertzel DFT", *Working Paper*, http://meyersanalytics.com/articles.htm
6. Press, William H., et al, "Numerical Receipts in C++: the Art of Scientific Computing", Cambridge Press, 2002.
7. Oppenheim, A, Schafer, R. and Buck, J., "Discrete Time Signal Processing", Prentice Hall, 1996, pp663-634
8. Proakis, J. and Manolakis, D. "Digital Signal Processing-Principles, Algorithms and Applications", Prentice Hall, 1996., pp480-481

Info on Dennis Meyers

Dennis Meyers (info@MeyersAnalytics.com.) has a doctorate in applied mathematics in engineering. He is a private trader, and president of Meyers Analytics (www.MeyersAnalytics.com). His firm specializes in Financial Engineering consulting for financial institutions and developing publicly available analytical software for traders.

Table 1 Walk Forward Out-Of-Sample Performance Summary for the TF 5-min Goertzel Strategy

TF-5 min bars 7/28/2010 - 8/31/2012. The input values *ncy, pntup, pntdn* are the values found from applying the filter to the in-sample section optimization runs.

Filter= Top 20 llt(Bottom 20 absolute value of llt values), maximum b1
osnp = Weekly Out-of-sample gross profit in $
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 $

In-Sample Dates			Out-Of-Sample Dates			osnp	NetEq	ollt	odd	ont	avosnp	ncy	pup	pdn	TF Cls
07/28/10	to	08/27/10	08/30/10	to	09/03/10	(2090)	(2090)	-1860	-4310	5	(418)	1	3	4.75	617.7
08/04/10	to	09/03/10	09/06/10	to	09/10/10	(2650)	(4740)	-550	-2030	4	(663)	1	3.5	5	610.5
08/11/10	to	09/10/10	09/13/10	to	09/17/10	2500	(2240)	0	0	2	1250	3	2.5	6	624.7
08/18/10	to	09/17/10	09/20/10	to	09/24/10	3410	1170	0	0	2	1705	3	2.5	6	644.5
08/25/10	to	09/24/10	09/27/10	to	10/01/10	(1000)	170	-740	-740	4	(250)	3	2	5.25	655.2
09/01/10	to	10/01/10	10/04/10	to	10/08/10	1250	1420	-20	-20	2	625	3	1	6	667.7
09/08/10	to	10/08/10	10/11/10	to	10/15/10	(440)	980	-750	-1710	6	(73)	3	2.5	6	679.9
09/15/10	to	10/15/10	10/18/10	to	10/22/10	(3750)	(2770)	-1390	-3900	4	(938)	1	1.5	6	679.1
09/22/10	to	10/22/10	10/25/10	to	10/29/10	1560	(1210)	0	0	5	312	10	4.5	4.5	679.2
09/29/10	to	10/29/10	11/01/10	to	11/05/10	450	(760)	-810	-1190	4	113	5	3.5	4.5	713.2
10/06/10	to	11/05/10	11/08/10	to	11/12/10	1320	560	0	0	3	440	10	5.5	6	695.2
10/13/10	to	11/12/10	11/15/10	to	11/19/10	2440	3000	-620	-620	4	610	10	4	6	700.6
10/20/10	to	11/19/10	11/22/10	to	11/26/10	(980)	2020	-1150	-1840	3	(327)	10	4.25	5.25	707.7
10/27/10	to	11/26/10	11/29/10	to	12/03/10	230	2250	-1370	-1950	4	58	10	2.25	6	731.7
11/03/10	to	12/03/10	12/06/10	to	12/10/10	1650	3900	-160	-160	4	413	10	3.75	6	752.9
11/10/10	to	12/10/10	12/13/10	to	12/17/10	(500)	3400	-610	-1200	4	(125)	10	4	5.75	755.5
11/17/10	to	12/17/10	12/20/10	to	12/24/10	1080	4480	-10	-10	2	540	5	1	4.25	766.5
11/24/10	to	12/24/10	12/27/10	to	12/31/10	200	4680	0	0	1	200	1	6	5.5	761.6
12/01/10	to	12/31/10	01/03/11	to	01/07/11	60	4740	-340	-340	3	20	1	1	6	764
12/08/10	to	01/07/11	01/10/11	to	01/14/11	(20)	4720	-980	-1070	4	(5)	3	1.5	5	785.2
12/15/10	to	01/14/11	01/17/11	to	01/21/11	(3400)	1320	0	0	1	(3400)	1	2.25	4.75	751.2
12/22/10	to	01/21/11	01/24/11	to	01/28/11	660	1980	0	0	1	660	1	3	5.25	754.1
12/29/10	to	01/28/11	01/31/11	to	02/04/11	2480	4460	-420	-420	2	1240	1	1.5	6	777.7
01/05/11	to	02/04/11	02/07/11	to	02/11/11	2250	6710	0	0	1	2250	1	1.5	6	800.1
01/12/11	to	02/11/11	02/14/11	to	02/18/11	1810	8520	0	0	1	1810	3	2	5.75	813.6
01/19/11	to	02/18/11	02/21/11	to	02/25/11	2460	10980	0	0	1	2460	5	2.25	5.75	801.1
01/26/11	to	02/25/11	02/28/11	to	03/04/11	520	11500	-330	-330	4	130	5	2.25	5.5	804.7
02/02/11	to	03/04/11	03/07/11	to	03/11/11	(740)	10760	-1380	-1660	3	(247)	1	2.25	6	781.4
02/09/11	to	03/11/11	03/14/11	to	03/18/11	(1680)	9080	-1280	-2320	3	(560)	1	3.5	4.5	774.2
02/16/11	to	03/18/11	03/21/11	to	03/25/11	1780	10860	-560	-560	2	890	1	3.5	5.5	802.8
02/23/11	to	03/25/11	03/28/11	to	04/01/11	1900	12760	-210	-210	2	950	1	3.5	5.5	827.3
03/02/11	to	04/01/11	04/04/11	to	04/08/11	(750)	12010	0	0	1	(750)	1	3.5	5.5	820.2
03/09/11	to	04/08/11	04/11/11	to	04/15/11	440	12450	0	0	1	440	1	5	5.25	815.7
03/16/11	to	04/15/11	04/18/11	to	04/22/11	3950	16400	0	0	1	3950	1	4.25	5.25	822.9
03/23/11	to	04/22/11	04/25/11	to	04/29/11	1480	17880	-450	-450	2	740	1	1	6	845.7
03/30/11	to	04/29/11	05/02/11	to	05/06/11	(1000)	16880	-650	-650	2	(500)	1	1	5	811.5
04/06/11	to	05/06/11	05/09/11	to	05/13/11	(3110)	13770	-1210	-2240	4	(778)	1	3.25	4.75	814.4
04/13/11	to	05/13/11	05/16/11	to	05/20/11	1220	14990	-1700	-1700	2	610	1	3.25	5.75	807.2
04/20/11	to	05/20/11	05/23/11	to	05/27/11	1600	16590	-490	-870	11	146	10	1.25	1.5	818.3

In-Sample Dates			Out-Of-Sample Dates			osnp	NetEq	ollt	odd	ont	avosnp	ncy	pup	pdn	TF Cls
04/27/11	to	05/27/11	05/30/11	to	06/03/11	(350)	16240	-920	-2170	4	(88)	5	5.5	3.25	789.4
05/04/11	to	06/03/11	06/06/11	to	06/10/11	(1470)	14770	-900	-2090	6	(245)	5	5.5	3.25	761
05/11/11	to	06/10/11	06/13/11	to	06/17/11	(840)	13930	-1530	-3360	6	(140)	3	6	3.25	762.9
05/18/11	to	06/17/11	06/20/11	to	06/24/11	(1190)	12740	-2470	-2470	4	(298)	3	6	4.75	779.3
05/25/11	to	06/24/11	06/27/11	to	07/01/11	(790)	11950	-1490	-1820	3	(263)	1	1	1.25	821.6
06/01/11	to	07/01/11	07/04/11	to	07/08/11	560	12510	-870	-870	6	93	10	1.5	3	836.5
06/08/11	to	07/08/11	07/11/11	to	07/15/11	2770	15280	-420	-540	6	462	5	1	1.25	813.3
06/15/11	to	07/15/11	07/18/11	to	07/22/11	4420	19700	-170	-170	5	884	5	1	5.75	823.8
06/22/11	to	07/22/11	07/25/11	to	07/29/11	(4100)	15600	-2070	-3970	6	(683)	5	1.25	4.75	780.5
06/29/11	to	07/29/11	08/01/11	to	08/05/11	(1310)	14290	-2430	-4320	14	(94)	10	1	1	698.5
07/06/11	to	08/05/11	08/08/11	to	08/12/11	2680	16970	-2410	-2410	7	383	3	5.25	2.25	681.9
07/13/11	to	08/12/11	08/15/11	to	08/19/11	5650	22620	-1440	-1490	16	353	5	5	1	639.1
07/20/11	to	08/19/11	08/22/11	to	08/26/11	4010	26630	-80	-80	3	1337	1	6	1.5	674.8
07/27/11	to	08/26/11	08/29/11	to	09/02/11	6920	33550	0	0	1	6920	1	6	1	665.6
08/03/11	to	09/02/11	09/05/11	to	09/09/11	7790	41340	0	0	2	3895	1	6	1	660.1
08/10/11	to	09/09/11	09/12/11	to	09/16/11	(4320)	37020	-270	-350	2	(2160)	1	6	4.25	701.7
08/17/11	to	09/16/11	09/19/11	to	09/23/11	3550	40570	-1680	-1680	3	1183	1	6	5.5	637.3
08/24/11	to	09/23/11	09/26/11	to	09/30/11	4840	45410	-790	-790	3	1613	1	6	2	630
08/31/11	to	09/30/11	10/03/11	to	10/07/11	5340	50750	0	0	2	2670	1	6	2	644.3
09/07/11	to	10/07/11	10/10/11	to	10/14/11	(1200)	49550	-980	-980	6	(200)	1	1.5	2	698.1
09/14/11	to	10/14/11	10/17/11	to	10/21/11	6630	56180	0	0	4	1658	1	1.75	2	699.4
09/21/11	to	10/21/11	10/24/11	to	10/28/11	4850	61030	-1230	-1230	3	1617	1	1	2.25	748
09/28/11	to	10/28/11	10/31/11	to	11/04/11	(1990)	59040	-3430	-3430	4	(498)	1	1	2.75	733.2
10/05/11	to	11/04/11	11/07/11	to	11/11/11	(1520)	57520	-1930	-4570	12	(127)	10	2.75	1.75	731.2
10/12/11	to	11/11/11	11/14/11	to	11/18/11	(2300)	55220	-1250	-2690	13	(177)	5	1	1	707
10/19/11	to	11/18/11	11/21/11	to	11/25/11	(900)	54320	-1670	-1700	6	(150)	5	1.75	1.25	652.5
10/26/11	to	11/25/11	11/28/11	to	12/02/11	5220	59540	-1010	-1340	8	653	5	1.75	1.25	721.4
11/02/11	to	12/02/11	12/05/11	to	12/09/11	(1720)	57820	-1460	-3750	11	(156)	5	1.5	1.25	735
11/09/11	to	12/09/11	12/12/11	to	12/16/11	4410	62230	-1480	-1920	6	735	3	2	1	710.6
11/16/11	to	12/16/11	12/19/11	to	12/23/11	250	62480	-1840	-2570	6	42	3	1.75	3.25	735.8
11/23/11	to	12/23/11	12/26/11	to	12/30/11	70	62550	0	0	1	70	3	3.75	2.75	731
11/30/11	to	12/30/11	01/02/12	to	01/06/12	(4050)	58500	-1290	-3610	6	(675)	3	5.25	1.75	739.8
12/07/11	to	01/06/12	01/09/12	to	01/13/12	(830)	57670	-790	-1150	4	(208)	3	2.75	3	755.8
12/14/11	to	01/13/12	01/16/12	to	01/20/12	1880	59550	0	0	2	940	3	1.75	5.5	775
12/21/11	to	01/20/12	01/23/12	to	01/27/12	1000	60550	-170	-170	2	500	3	1.5	6	788.4
12/28/11	to	01/27/12	01/30/12	to	02/03/12	2650	63200	-300	-540	2	1325	3	1.75	6	820.1
01/04/12	to	02/03/12	02/06/12	to	02/10/12	(1410)	61790	-750	-750	2	(705)	1	1.25	6	805.5
01/11/12	to	02/10/12	02/13/12	to	02/17/12	690	62480	-370	-370	2	345	1	1	6	819.8
01/18/12	to	02/17/12	02/20/12	to	02/24/12	490	62970	0	0	2	245	1	1.25	6	817.8
01/25/12	to	02/24/12	02/27/12	to	03/02/12	(2190)	60780	0	0	1	(2190)	1	1.25	6	793.1
02/01/12	to	03/02/12	03/05/12	to	03/09/12	330	61110	-650	-1440	15	22	10	1	1	810.4
02/08/12	to	03/09/12	03/12/12	to	03/16/12	650	61760	-1090	-1170	4	163	1	3.75	1.25	822.7
02/15/12	to	03/16/12	03/19/12	to	03/23/12	2300	64060	-890	-940	5	460	10	2.75	1.75	824.4
02/22/12	to	03/23/12	03/26/12	to	03/30/12	500	64560	-1150	-1300	9	56	10	2.75	1.75	823.7
02/29/12	to	03/30/12	04/02/12	to	04/06/12	1240	65800	0	0	1	1240	1	6	1	809.1
03/07/12	to	04/06/12	04/09/12	to	04/13/12	3970	69770	-1230	-1230	3	1323	1	6	2	789.2
03/14/12	to	04/13/12	04/16/12	to	04/20/12	1410	71180	-940	-940	8	176	10	3.25	1.5	797.3
03/21/12	to	04/20/12	04/23/12	to	04/27/12	(640)	70540	-990	-2610	11	(58)	10	3.75	1	819.8
03/28/12	to	04/27/12	04/30/12	to	05/04/12	420	70960	-1220	-1220	8	53	10	3.75	5.5	782.6
04/04/12	to	05/04/12	05/07/12	to	05/11/12	(1860)	69100	-620	-2830	9	(207)	5	1	2.25	784.1
04/11/12	to	05/11/12	05/14/12	to	05/18/12	(70)	69030	-1430	-1900	4	(18)	3	3	4.5	740.3
04/18/12	to	05/18/12	05/21/12	to	05/25/12	2640	71670	-500	-840	11	240	10	2.75	2	761.1
04/25/12	to	05/25/12	05/28/12	to	06/01/12	(340)	71330	-1190	-2140	6	(57)	3	3.25	2.25	732.6

In-Sample Dates			Out-Of-Sample Dates			osnp	NetEq	ollt	odd	ont	avosnp	ncy	pup	pdn	TF Cls
05/02/12	to	06/01/12	06/04/12	to	06/08/12	(1840)	69490	0	0	2	(920)	1	6	1	766.4
05/09/12	to	06/08/12	06/11/12	to	06/15/12	420	69910	-610	-940	9	47	10	5.25	3.5	766.9
05/16/12	to	06/15/12	06/18/12	to	06/22/12	(50)	69860	0	0	1	(50)	1	5.75	3.25	769
05/23/12	to	06/22/12	06/25/12	to	06/29/12	2550	72410	-240	-410	9	283	10	1.5	3.5	795.4
05/30/12	to	06/29/12	07/02/12	to	07/06/12	3380	75790	-550	-550	3	1127	10	1	2	804.6
06/06/12	to	07/06/12	07/09/12	to	07/13/12	(990)	74800	-1150	-2450	11	(90)	10	1.5	1	798.6
06/13/12	to	07/13/12	07/16/12	to	07/20/12	(980)	73820	-1340	-1610	5	(196)	10	3	4	788.7
06/20/12	to	07/20/12	07/23/12	to	07/27/12	(1000)	72820	-1660	-2870	7	(143)	5	2.5	2.75	796.6
06/27/12	to	07/27/12	07/30/12	to	08/03/12	5460	78280	0	0	2	2730	5	4	3.25	786.7
07/04/12	to	08/03/12	08/06/12	to	08/10/12	(1170)	77110	-950	-2090	4	(293)	1	3	6	799
07/11/12	to	08/10/12	08/13/12	to	08/17/12	1770	78880	0	0	1	1770	1	4	6	816.5
07/18/12	to	08/17/12	08/20/12	to	08/24/12	2390	81270	0	0	3	797	10	5.5	5	807.1
07/25/12	to	08/24/12	08/27/12	to	08/31/12	(460)	80810	-350	-720	5	(92)	3	4.25	5.75	811.1

osnp = Weekly Out-of-sample gross profit in $
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 In-Sample Strategy Inputs Each Week To the Out-Of-Sample TF 5min Bar Prices 9/03/10 – 8/31/12

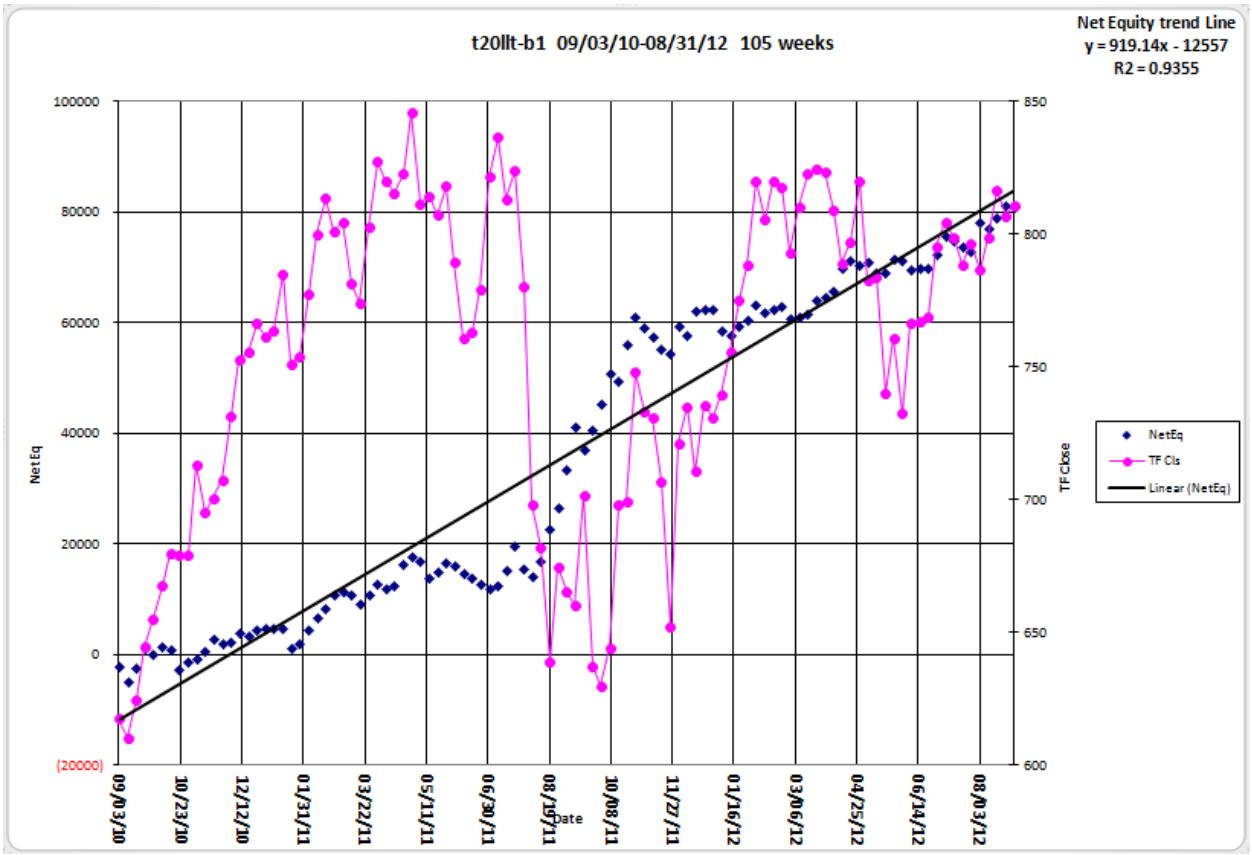

Figure 2 TF GZ System 5 minute bar chart from 8/14/12-8/31/12

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

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	TF5GZnoxt729	20100903	20120831	#105	AnyTnp				Boot Ave=7551		Boot Std=19886		f1922							c=$ 0
2	Filter-Metric	tOnp	mOnp	aOnp	aOTrd	aO#T	%P	t	std	llw	eqDD	lr	#	eqTrn	eqR2	Blw	BE	ndd	tOnpNet	Prob
3	t20llt-b1	80810	450	770	170.8	4.5	59	3.17	2484.6	-4320	-6710	5	105	919	94	11	41.7	8.3	80810	8.36E-05
4	t10llt-b1	75930	340	723	162.9	4.4	59	3.09	2394.6	-4320	-5640	5	105	828	94	10	43.9	7.4	75930	2.21E-04
5	t20awl-b2	75130	630	716	194.1	3.7	61	2.69	2728.8	-5080	-7700	4	105	1010	90	19	58.2	10.2	75130	2.58E-04
6	t20em3-mpft	71850	390	684	177.8	3.8	57	2.82	2489.7	-3540	-9040	6	105	936	89	21	53	12.6	71850	4.76E-04

The WFMPE Filter Output Columns are defined as follows

Row 1 TF5Gznoxt729 is the PWFO Stub, OOS File Start Date(6/27/08), OOS File End Date(6/18/10), Number of weeks(#105) Boot ave =average of bootstrap random picks. Boot Std= standard deviation of bootstrap random picks. f=number of different filters examined. C= slippage and round trip trade cost(c=$0).

Filter-Metric = The filter that was run. For example, t20llt-b1
Metric = The PWFO performance Metric=b1 (In-sample Section Trade Equity 2nd Order Polynomial Trend Line Slope Coefficient).

This Filter t20llt-b1 filter produced the following average 105 week statistics on this line(row 3).

tOnp = Total out-of-sample(oos) net profit for these 105 weeks.

mOsp = median oos net profit for the 105 weeks

aOsp = Average oos net profit for the 105 weeks

aOTrd = Average oos profit per trade

aO#T = Average number of oos trades per week

%P = The percentage of oos weeks that were profitable

t = The student t statistic for the 105 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 105 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.

eqTrn = 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.