Trading 1Min Bar Euro Futures Using
The Nth order Fixed Memory Polynomial Velocity Strategy

August 1, 2011 –May 26, 2017

Working Paper May 2017 Copyright © 2017 Dennis Meyers

Disclaimer

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

In previous working papers at http://www.meyersanalytics.com/papers.php we showed how the application of a price curve generated by the Nth Order Fixed Memory Polynomial Velocity could be used to develop a strategy to buy and sell futures intraday. The reasoning behind this type of system was to only trade when the price trend velocity was above a certain threshold. Many times, prices meander around without any notable trend and this is considered noise. During these times, we do not wish to trade because of the cost of whipsaw losses that would occur from this type of price action. When a price trend finally starts, the velocity of that price trend moves above a minimum threshold noise value. Thus, the velocity system would only issue a trade when certain velocity thresholds above “noise” levels are crossed.

The velocity system that we will use here to trade the Euro futures contract is called the nth Order Adaptive Polynomial Velocity Strategy. The word “Adaptive” is used because the polynomial inputs change over time, adapting to the changing trading patterns of the Euro futures contract. The nth Order Adaptive Polynomial Velocity Strategy has a number of unknown inputs that we must determine before we can use this strategy to trade. These unknown inputs are the polynomial order or degree, the optimum number of lookback prices we need to determine the co-efficients of the polynomial and finally the velocity thresholds. Here we will use Walk Forward Optimization and out-of-sample testing to determine the “best” polynomial inputs as well as how these inputs should change over time. We will use the nth Order Adaptive Polynomial Velocity System to trade the Euro futures contract on an intraday basis using one minute bar price data. To test this strategy we will use one minute bar prices of the Euro futures contract(EC) traded on the CME/Globex from August 1, 2011 to May 26, 2017.

The n^{th} Order Fixed Memory System Defined

The least squares forecast n^{th} order fixed memory polynomial velocity is constructed by solving for the coefficients $\beta_0, \beta_1, \beta_2, \beta_3 \ldots \beta_n$ for the discrete orthogonal Legendre polynomials each day using the last **N bars** of closing prices and the equation for β_j shown in the **“Math”** appendix at the end of this working paper. Then **Velocity(T+1)** is constructed from the equation shown in the “Math” appendix and plotted under the price chart.

Due to polynomial mathematics, the Velocity of the 2^{nd} , 3^{rd} and 4^{th} order degree polynomial curve changes faster than it’s corresponding first order degree polynomial straight line velocity. Whether higher order polynomial velocities are an advantage or not, will be determined by the computer when we use a walk forward optimization technique described below.

At each bar we calculate the n^{th} order degree(1^{st} through 4^{th}) fixed memory polynomial velocity from the formulas in the “Math” appendix. As will be shown below, walk forward optimization will determine the degree for the nth order polynomial velocity, the number of lookback prices, N, needed to compute the polynomial coefficients and the threshold amounts vup and vdn. When the nth order degree velocity is greater than the threshold amount ***vup*** we will go long. When the velocity is less than the threshold amount ***-vdn*** we will go short.

Buy Rule:

IF Velocity is greater than the threshold amount ***vup*** then buy at the market.

Sell Rule:

IF Velocity is less than the threshold amount ***-vdn*** then sell at the market.

Intraday Bars Exit Rule:
Close the position at 1500 CST. Trading volume on Globex dies down substantially by 1500 CST. No trades will be carried overnight.

Intraday Bars First Trade of Day Entry Rule:

Ignore all trade signals before **7:00**am. For the Buy and Sell rules above we have included a first trade of the day entry rule. Trading in the EC futures has changed a lot because of 24hr Globex trading. Trading starts a lot earlier in the morning right after midnight when Asia and then later Europe opens and then dies down. The EC volume starts to pick up again around 7am CST when US trading starts to kick in so we will have the strategy start trading at 7am.

Discussion of Euro Prices

The Euro (EC) is traded on Globex. On Globex the EC is traded on a 23 hour basis. We have restricted our study to only trading the EC during the 7:00 to 1500 hours.

Testing The Polynomial Velocity System Using Walk Forward Optimization

There will be four strategy parameters to determine:

1. **degree**, degree=1 for straight line velocity, degree=2 for 2nd order velocity, etc.
2. ***N***, is the number of lookback bars of prices to calculate the **velocity.**
3. ***vup***, the threshold amount that velocity has to be greater than to issue a buy signal
4. ***vdn***, the threshold amount that velocity has to be less than to issue a sell signal

To test this system, we will use one minute bar prices of the Euro (EC) futures contract traded on the CME/Globex and known by the symbol EC from August 1, 2011 to May 26, 2017.

We will test this strategy with the above EC 1 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 the nth Order Polynomial Velocity Strategy on the EC 1min data from August 1, 2011 to May 26, 2017. The PWFO will breakup and create 30 day calendar in-sample sections along with their corresponding one calendar week out-of-sample sections from the 304 weeks of EC (see Walk Forward Testing below) creating 304 out-of-sample weeks.

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

Whenever we do a TS optimization on many 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 is 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 an ***in-sample*** or ***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 ***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” or “Data Mining” (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, that is price data they have not been optimized on.

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

The PWFO is a TS ***add-in*** that breaks up the TS optimization run into many user selectable in-sample and out-of-sample sections. The PWFO prints out the in-sample sample performance metrics **and the out-of-sample performance results**, on one line, for each case or set of in-sample 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 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 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 **http://meyersanalytics.com/Walk-Forward-Optimization.html** for a listing of these performance metrics).

For our computer run we will have the PWFO breakup the 304 weeks of EC one minute bar price data into 304 in-sample and 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. As an example the first in-sample section would be from 6/30/2011 to 7/29/2011 and the out-of-sample section would be the week following from 8/1/2011 to 8/5/2011.(our in-sample and out-of-sample sections always end on a Friday). We would then move everything ahead a week and the 2nd in-sample section would be from 7/7/2011 to 8/5/2011 and the week following out-of-sample section would be from 8/8/2011 to 8/12/2011. Etc.

The PWFO 304 in-sample/out-of-sample section dates are shown in **Table 1** below. We will then use another software product called the Walk Forward Performance Metric Explorer (WFME) on each of the 304 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 ***(degree, N, vup, vdn)*** that will be used on the out-of-sample data. Detailed information about the WFME can be found at **http://meyersanalytics.com/wfme.html**

For the in-sample data we will run the TradeStation optimization engine on the 304 weeks of EC 1 min bars with the following ranges for the nth order fading memory polynomial velocity strategy input variables.

1. **pw=degree from 1 to 4**
2. **N from 20 to 70 in steps of 10.**
3. **vup from 0.25 to 3.5 steps of 0.25**
4. **vdn from 0.25 to 3.5 in steps of 0.25**
5. **Mult = 15000, iNorm=1 (See Appendix 3, the Normalization Multiplier)**

This will produce 4704 different cases or combinations of the input parameters for each of the 304 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 in-sample optimization run. For instance, a ***filter*** might be all cases that have a profit factor (PF) greater than 1 and less than 3. For the number of cases left, we might select the cases that had the best percent profit. This procedure would leave you with one case in the in-sample section output 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 and 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 strategy 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 in-sample 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 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 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 strategy optimization, neural net optimizations or combinatorial searches with no out-of-sample testing cause loses when traded in real time by mistaking chance fluctuations for genuine effects. Unfortunately it is human nature to extrapolate past performance to project future trading results and thus results from the chance fluctuations of 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 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 or just three times? 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 strategy performance filter 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 equity and its path ranges 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 in-sample section. (Please see **http://meyersanalytics.com/Walk-Forward-Optimization.html** for a listing of these performance metrics). 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 out-of-sample sections. The PWFO produces a total of 32 different performance metrics in the in-sample section. If we have 4704 different input variations or cases then the in-sample section consists of 32 columns of performance metrics for each of the 4704 different strategy inputs 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 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 a combination ***filter*** that is used in this paper with good out-of-sample results. High profit factors (**PF**) in the in-sample section usually mean poor performance in the out-of-sample-section. This is a kind of reversion to the mean. So in the in-sample section we eliminate all strategy input rows that have a **PF>4**. **R2** is defined as the in-sample trade equity regression trend line coefficient of correlation $\mathbf{r^2}$. $\mathbf{r^2}$ is a measure of how well a straight line fits the equity curve generated by a set of in-sample strategy inputs. High **r2** values in the in-sample section usually mean poor performance in the out-of-sample-section. This is a kind of reversion to the mean and a measure of how well the price noise is being fitted in the IS section. So, in the in-sample section we eliminate all strategy input rows that have a **r2>90**. After using the **pf-r2** filter, as described, there can still be 100's of rows left in the in-sample section. Few traders can stay with a strategy that has many losing trades in a row (**lr**). For this filter we will limit the number losing trades in a row in the 30 day IS period to 5 or less (**lr<5).** The PWFO generates the metric **tLb.** This metric is the **Total Losing Bars in the losing trades in the in-sample section.** Let us choose the 20 rows in the in-sample section that contain the lowest(bottom) number of **tLb** from the rows that are left from the **PF-R2-LR** elimination. 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 minimum metric **eq2V** in the in-sample section**. eq2V is the Velocity of a 2nd Order Polynomial Line fit to the equity curve evaluated at the end of the equity curve. T**his would produce a filter named **b20tLb|pf<4|lr<5|r2<90-eq2V.** This in-sample 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 **b20tLb|pf<4|lr<5|r2<90-eq2V** filter finds the strategy inputs parameters in each of the 304 in-sample sections and applies these inputs to each of the 304 out-of-sample sections. Using the filter in-sample strategy inputs on the 304 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 filter 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. A total of 43200 different metric filters were examined. We chose **Row 3** because it had a lower **BE**, **BLW** and **Dev^2** and a higher **KTau** and **eqR2** along with better statistics than the rows below it. More on this below and on how that number of filters combinations effect the probability that the filter chosen was or was not due to chance

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(toNP) over all out-of-sample weeks for a given filter like above. A mirror filter is created. However, instead of picking an out-of-sample net profit(OSNP) from a row that the filter picks, the mirror filter picks a ***random*** row's OSNP in each of the 304 PWFO files. Suppose we repeat this random row section 5000 times. Each of the 5000 mirror filters will choose a random row's OSNP of their own in each of the 304 PWFO files. At the end, each of the 5000 mirror filters will have 304 ***random*** OSNP's picked from the rows of the 304 PWFO files. The sum of the 304 random OSNP picks for each of the 5000 mirror filters will generate a random total out-of-sample net profit(toNP). The average and standard deviation of the 5000 mirror different random toNPs will allow us to calculate the chance probability for each 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 304 out-of-sample files of **($113.3)** with a bootstrap standard deviation of **$65.9**. (Side Note. The average is the average per out-of-sample period(weekly). So, the average for the random selection would be the random (Average Random toNP/304) and the average net weekly for the filter would be the filter toNP/ (# of OOS) periods traded or **37089/158=234.7**. The probability of obtaining our filters average weekly net profit **of 234.7** is **$6.32\text{x}10^{-8}$** which is **5.28** 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 **$234.7** is
[1-(1- **$6.32\text{x}10^{-8})^{43200}$** ~ **43200**x **$6.32\text{x}10^{-8}$** = **0.0027** where **43200** is the total number of different filters we looked at in this run. This number is much less than one so it is improbable that our result was due to pure chance.

Results

Table 1 below presents a table of the 304 in-sample and out-of-sample windows, the selected optimum parameters and the weekly out-of-sample results using the filter described above.

Figure 1 presents a graph of the equity and net equity curves generated by using the filter on the 304 weeks ending 8/5/11 to 5/26/17. The equity curves are plotted from the Equity and Net Equity columns in Table 1. Plotted on the equity curves are 2nd Order Polynomial fits. The blue line is the equity curve without commissions and the red dots on the blue line are new highs in equity. The brown line is the net equity curve with commissions and the green dots are the new highs in net equity. The grey line is the weekly EC prices superimposed on the equity chart.

Figure 2 Walk Forward Out-Of-Sample Performance for EC Nth Order Fixed Memory Polynomial Velocity System 1 minute bar chart of EC from 3/10/2016

Figure 3 Partial output of the Walk Forward Metric Performance Explorer (WFME)
Run on the 304 PWFO files of the EC 1min bars Nth Order Fixed Memory Polynomial Velocity

Discussion of Strategy Performance

In Figure 3 Row 3 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 29 weeks for this filter. This means that 29 weeks was the longest time that the equity for this strategy failed to make a new equity high in the 304 out-of-sample weeks. For this strategy, the **%P** (% of oos periods that are positive) was 61%, and the **%Wtr** (The % of all oos trades that are positive) was 56%.

To see the effect of walk forward analysis, look at **Table 1**. Notice how the input parameters ***pw, N, vup and vdn*** 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 ***degree*** changes from a straight-line velocity with ***degree=1*** to a 2nd, 3rd and 4th order velocity with ***degree= 2, 3 and 4***. The 3rd and 4th order velocities, due to the higher order components, change much faster than the straight-line velocity. When the data gets very noisy with a lot of spurious price movements, it's better to have the velocity change slower filtering out the noisy data. During other times when the noise level is not as much it is better to have the velocity break its ***vup*** and ***vdn*** barriers faster to get onboard a trend faster. This is what the filter is doing. When there is a lot of noise in the in-sample section it switches to the 1st or 2nd order curve velocity. When the noise level is lower in the in-sample section, it switches to the faster changing 3rd or 4th order curve velocity.

In Figure 1, which presents a graph of the equity curve using the filter on the 304 weeks of out-of-sample data, notice how the equity curve follows the 2nd order polynomial trend line with an R^2 of 0.987. This R^2 dropped to 0.984 for the net equity curve.

In observing Table 1 we can see that this strategy and filter made trades from a low of no trades/week to a high of 9 trades/week with an average of 2.1 trades/week on the weeks it did trade. The strategy seemed to wait for really strong trends and then initiate a buy or sell. There were many weeks that had no trades. Out of the 304 out-of-sample weeks the filter only traded 158 of those weeks or 52% of the time with 56% of all trades profitable and 61% of all weeks profitable. In observing the Equity Curve plot in Figure 1 we can see that the equity did quite well in both big up and down moves of the EC. In observing the chart from 5/9/14 to 3/13/15 where EC declined from 1.4027 to 1.0613 we can see the strategy did well on this unusually large Euro decline making $7200 and only trading 17 weeks out of those 45 weeks. . After the Brexit vote on 6/23/2016, the strategy did well in the volatile market from Friday 6/24 to Friday 7/1 only losing -$556 on 3 trades that week.

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.

Figure 1 Graph of Net Equity Curve Applying the Walk Forward Filter Each Week On EC 1min Bar Prices 08/05/11 – 5/26/17

Note: The blue line is the equity curve without commissions and the red dots on the blue line are new highs in equity. The brown line is the equity curve with commissions and the green dots are the new highs in net equity The grey line is the EC Weekly Closing prices superimposed on the Equity Chart.

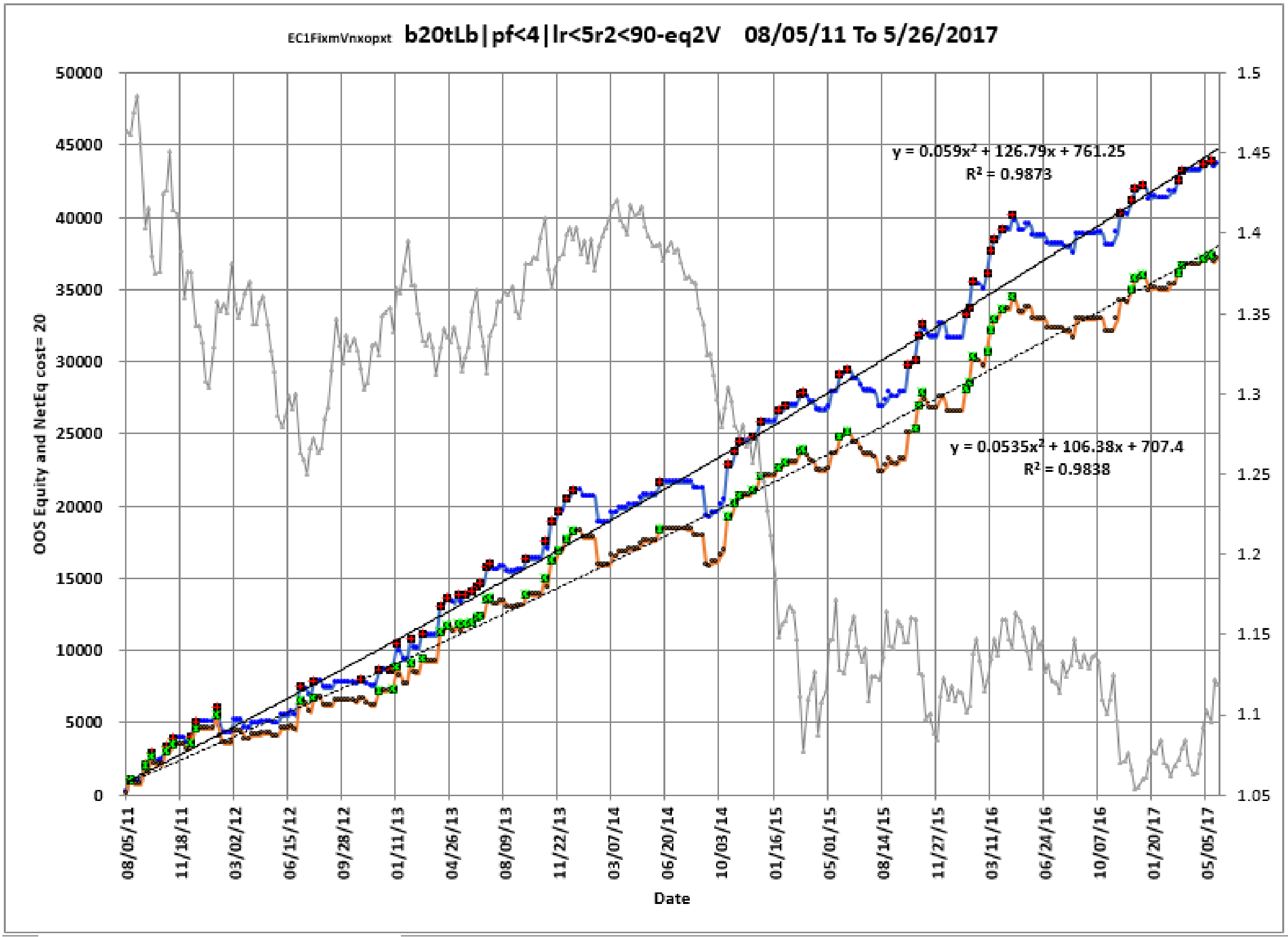
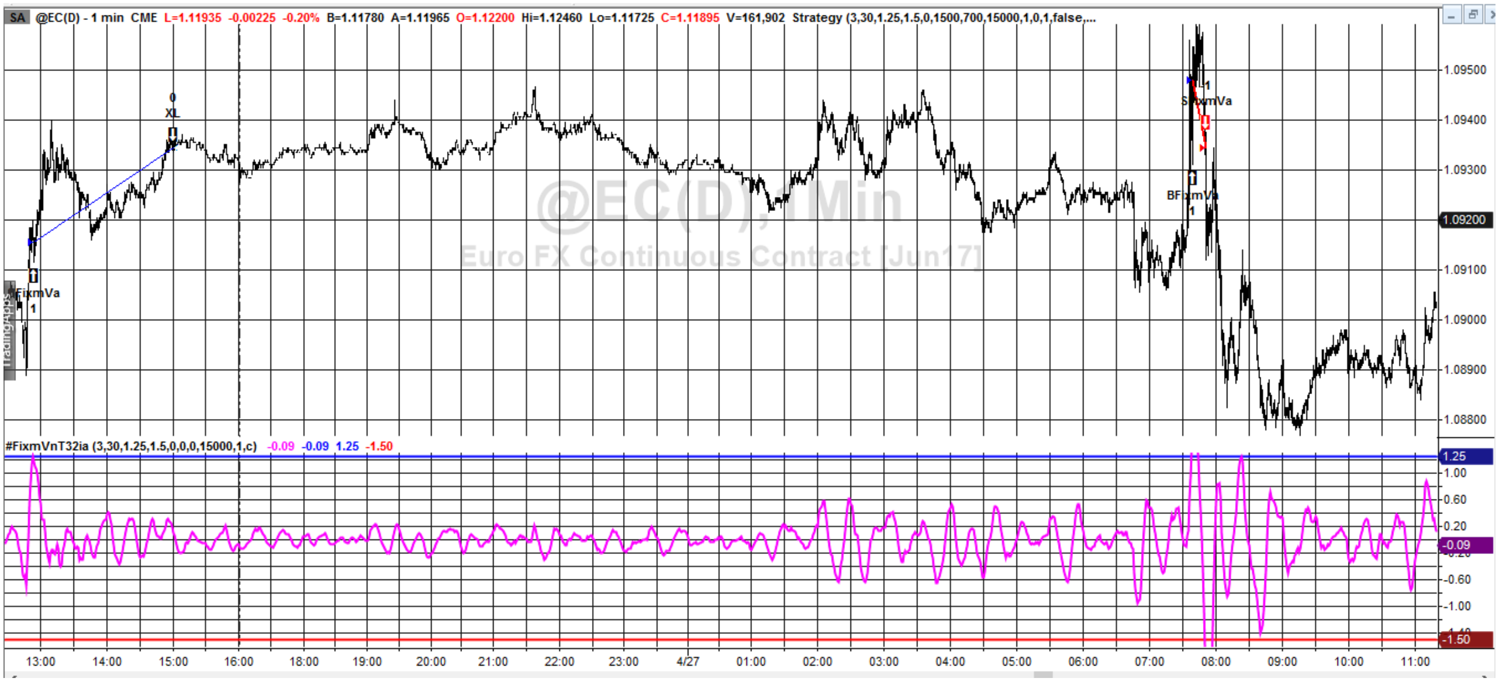


Figure 2 Walk Forward Out-Of-Sample Performance Summary for Nth Order Fixed Memory Polynomial Velocity System EC 1 minute bar chart from 4/26/17 4/27/17



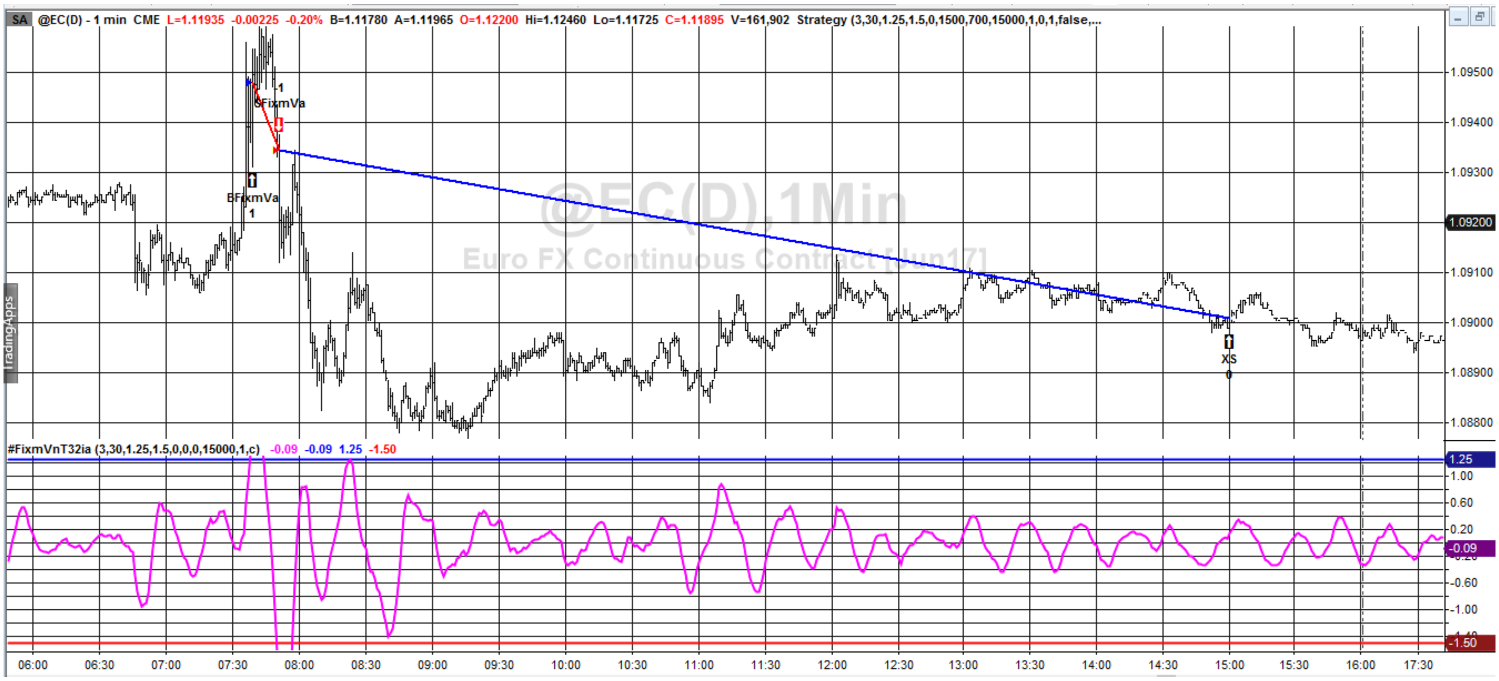
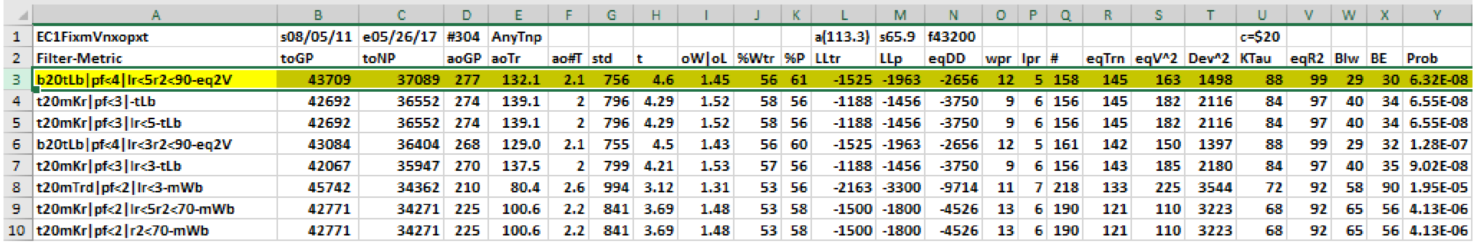


Figure 3_ Partial output of the Walk Forward Metric Performance Explorer (WFME)
EC1 min bars Nth Order Fixed Memory Velocity System



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	EC1FixmVnxopxt	s08/05/11	e05/26/17	#304	AnyTnp							a(113.3)	s65.9	f43200							c=$20				
2	Filter-Metric	toGP	toNP	aoGP	aoTr	ao#T	std	t	oW|oL	%Wtr	%P	LLtr	LLp	eqDD	wpr	lpr	#	eqTrn	eqV^2	Dev^2	KTau	eqR2	Blw	BE	Prob
3	b20tLb|pf<4|lr<5r2<90-eq2V	43709	37089	277	132.1	2.1	756	4.6	1.45	56	61	-1525	-1963	-2656	12	5	158	145	163	1498	88	99	29	30	6.32E-08
4	t20mKr|pf<3|-tLb	42692	36552	274	139.1	2	796	4.29	1.52	58	56	-1188	-1456	-3750	9	6	156	145	182	2116	84	97	40	34	6.55E-08
5	t20mKr|pf<3|lr<5-tLb	42692	36552	274	139.1	2	796	4.29	1.52	58	56	-1188	-1456	-3750	9	6	156	145	182	2116	84	97	40	34	6.55E-08
6	b20tLb|pf<4|lr<3r2<90-eq2V	43084	36404	268	129.0	2.1	755	4.5	1.43	56	60	-1525	-1963	-2656	12	5	161	142	150	1397	88	99	29	32	1.28E-07
7	t20mKr|pf<3|lr<3-tLb	42067	35947	270	137.5	2	799	4.21	1.53	57	56	-1188	-1456	-3750	9	6	156	143	185	2180	84	97	40	35	9.02E-08
8	t20mTrd|pf<2|lr<3-mWb	45742	34362	210	80.4	2.6	994	3.12	1.31	53	56	-2163	-3300	-9714	11	7	218	133	225	3544	72	92	58	90	1.95E-05
9	t20mKr|pf<2|lr<5r2<70-mWb	42771	34271	225	100.6	2.2	841	3.69	1.48	53	58	-1500	-1800	-4526	13	6	190	121	110	3223	68	92	65	56	4.13E-06
10	t20mKr|pf<2|r2<70-mWb	42771	34271	225	100.6	2.2	841	3.69	1.48	53	58	-1500	-1800	-4526	13	6	190	121	110	3223	68	92	65	56	4.13E-06

The WFME Filter Output Columns are defined as follows:

Row 1 EC1FixmVnxopxt is the strategy abbreviation, First OOS Week End Date (8/5/11), Last OOS Week End Date (5/26/17), **Number of weeks** (#304), **a**=weekly average of bootstrap random picks, **s**= weekly standard deviation of bootstrap random picks. **f**=number of different filters examined. **c**= slippage and round trip trade cost(c=$20).

Filter = The filter that was run. Row3 filter **b20tLb|pf<4|lr<5r2<90-eq2V**

The b20tLb|pf<4|lr<5|r2<90-eq2V filter produced the following average 304 week statistics on Row 3.

toGP = Total out-of-sample(oos) gross profit for these 304 weeks.

toNP = Total out-of-sample gross profit(toGP) minus the total trade cost.
toNP=toGP **–** (Number of trade weeks)*ao#T*Cost.

aoGP = Average oos gross profit for the 304 weeks

aoTr = Average oos profit per trade

ao#T = Average number of oos trades per week

std = standard deviation of weekly oos gross profit

t = The student t statistic for the 304 weekly oos profits. The higher the t statistic the higher the probability that this result was not due to pure chance

oW|oL = The average oos winning trade divided by the average oos losing trade

%Wtr = percent of all oos trades that were winning trades

%P = percent of all oos periods that were profitable.

LLtr = The largest losing oos trade in all oos periods,

LLp = The largest losing oos period(week)

eqDD = The oos equity drawdown

wpr = The largest number of winning oos periods (weeks) in a row.

lpr = The largest number of losing oos periods (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 and thus there are no trades for that oos period.

eqTrn = The straight line trend of the oos gross profit equity curve in $/week.

eqV^2 = The ending velocity of 2^{nd} order polynomial that is fit to the oos equity curve

eqR2 = The correlation coefficient(r^2) of a straight line fit to the oos equity curve

Dev^2 = A measure of equity curve smoothness. The square root of the average [(equity curve minus a straight line)^2]

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 98% probability that your oos equity is above zero.

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

Table 1 Walk Forward Out-Of-Sample Performance Summary for **EC Nth Order Fixed Memory Polynomial Velocity Strategy**

EC-1 min bars 8/5/2011 - 07/29/2016. The input values *degree(pw), N, vup, vdn* are the values found from applying the filter to the in-sample section optimization runs.

Filter= b20tLb|pf<4|lr<5|r2<90-eq2V

osnp = Weekly Out-of-sample gross profit in $
NOnp$20 = Weekly Out-Of-Sample Net Profit in $ = **osnp-ont*20**.
ont = The number of trades in the out-of-sample week.
ollt = The largest losing trade in the out-of-sample section in $.
odd = The drawdown in the out-of-sample section in $.
Equity = Running Sum of weekly out-of-sample gross profits $
NetEq = running sum of the weekly out-of-sample net profits in $
pw= degree, degree=1 for straight line velocity, degree=2 for 2nd order velocity, etc.
N = N the lookback period
vup, the threshold amount that velocity has to be greater than to issue a buy signal
vdn, the threshold amount that velocity has to be less than to issue a sell signal
Note: Blank rows indicate that no out-of-sample trades were made that week

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
06/30/11	to	07/29/11	08/01/11	to	08/05/11	175	95	4	-725	-1063	175	95	4	30	3.25	2.75
07/07/11	to	08/05/11	08/08/11	to	08/12/11	925	905	1	0	0	1100	1000	3	50	3.25	2.75
07/14/11	to	08/12/11	08/15/11	to	08/19/11	(225)	(245)	1	-225	-225	875	755	3	50	3.25	2.75
07/21/11	to	08/19/11	08/22/11	to	08/26/11							755	3	50	3.25	2.75
07/28/11	to	08/26/11	08/29/11	to	09/02/11							755	3	50	3.25	2.75
08/04/11	to	09/02/11	09/05/11	to	09/09/11	1263	1243	1	0	0	2138	1998	3	50	3.25	2.75
08/11/11	to	09/09/11	09/12/11	to	09/16/11	(438)	(458)	1	-438	-438	1700	1540	1	60	2.25	2
08/18/11	to	09/16/11	09/19/11	to	09/23/11	1150	1110	2	0	0	2850	2650	1	40	1.75	2.5
08/25/11	to	09/23/11	09/26/11	to	09/30/11	(550)	(570)	1	-550	-550	2300	2080	2	70	2	2.75
09/01/11	to	09/30/11	10/03/11	to	10/07/11	50	(10)	3	-1525	-1525	2350	2070	1	30	2.25	3.5
09/08/11	to	10/07/11	10/10/11	to	10/14/11	63	43	1	0	0	2413	2113	2	50	2.5	3.5
09/15/11	to	10/14/11	10/17/11	to	10/21/11	975	895	4	0	0	3388	3008	2	40	1.75	2.25
09/22/11	to	10/21/11	10/24/11	to	10/28/11							3008	3	20	3.25	3.25
09/29/11	to	10/28/11	10/31/11	to	11/04/11	538	518	1	0	0	3926	3526	1	40	2.5	3.25
10/06/11	to	11/04/11	11/07/11	to	11/11/11							3526	2	50	2.75	3.5
10/13/11	to	11/11/11	11/14/11	to	11/18/11							3526	1	40	2.5	2.75
10/20/11	to	11/18/11	11/21/11	to	11/25/11							3526	2	70	2.75	2.5
10/27/11	to	11/25/11	11/28/11	to	12/02/11	(375)	(415)	2	-413	-413	3551	3111	1	70	2.25	2
11/03/11	to	12/02/11	12/05/11	to	12/09/11	488	448	2	-13	-13	4039	3559	2	40	2.75	2.5
11/10/11	to	12/09/11	12/12/11	to	12/16/11	1038	1018	1	0	0	5077	4577	2	50	3.5	2
11/17/11	to	12/16/11	12/19/11	to	12/23/11							4577	2	40	3	3.25
11/24/11	to	12/23/11	12/26/11	to	12/30/11							4577	2	40	3	3.25
12/01/11	to	12/30/11	01/02/12	to	01/06/12							4577	3	50	2.5	2.25
12/08/11	to	01/06/12	01/09/12	to	01/13/12							4577	3	30	2.5	2.75
12/15/11	to	01/13/12	01/16/12	to	01/20/12							4577	4	20	2	2.25
12/22/11	to	01/20/12	01/23/12	to	01/27/12	988	908	4	-913	-913	6065	5485	1	60	1.5	1.25
12/29/11	to	01/27/12	01/30/12	to	02/03/12	(1388)	(1488)	5	-788	-1388	4677	3997	4	40	2	2.25
01/05/12	to	02/03/12	02/06/12	to	02/10/12	(363)	(423)	3	-150	-363	4314	3574	4	50	2.75	2.25
01/12/12	to	02/10/12	02/13/12	to	02/17/12							3574	3	40	2.75	2.75

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
01/19/12	to	02/17/12	02/20/12	to	02/24/12							3574	3	40	2.75	2.75
01/26/12	to	02/24/12	02/27/12	to	03/02/12	825	805	1	0	0	5139	4379	3	40	2.75	2.75
02/02/12	to	03/02/12	03/05/12	to	03/09/12							4379	4	60	3	3
02/09/12	to	03/09/12	03/12/12	to	03/16/12							4379	4	60	3	3
02/16/12	to	03/16/12	03/19/12	to	03/23/12	(488)	(548)	3	-488	-538	4651	3831	4	60	2	2
02/23/12	to	03/23/12	03/26/12	to	03/30/12							3831	4	70	2.25	2.25
03/01/12	to	03/30/12	04/02/12	to	04/06/12	288	268	1	0	0	4939	4099	1	60	1.75	1.75
03/08/12	to	04/06/12	04/09/12	to	04/13/12							4099	1	60	1.75	1.75
03/15/12	to	04/13/12	04/16/12	to	04/20/12							4099	4	50	2	2
03/22/12	to	04/20/12	04/23/12	to	04/27/12	138	118	1	0	0	5077	4217	4	20	1.75	2.5
03/29/12	to	04/27/12	04/30/12	to	05/04/12							4217	4	20	1.75	2.5
04/05/12	to	05/04/12	05/07/12	to	05/11/12							4217	4	20	1.5	2
04/12/12	to	05/11/12	05/14/12	to	05/18/12	(138)	(178)	2	-275	-275	4939	4039	2	20	1.5	3
04/19/12	to	05/18/12	05/21/12	to	05/25/12							4039	4	20	2.5	2.25
04/26/12	to	05/25/12	05/28/12	to	06/01/12	563	543	1	0	0	5502	4582	4	20	2.5	2.25
05/03/12	to	06/01/12	06/04/12	to	06/08/12							4582	4	20	2.5	2.25
05/10/12	to	06/08/12	06/11/12	to	06/15/12							4582	4	20	2.5	2.25
05/17/12	to	06/15/12	06/18/12	to	06/22/12	225	165	3	-388	-388	5727	4747	4	40	2	1.75
05/24/12	to	06/22/12	06/25/12	to	06/29/12	(275)	(295)	1	-275	-275	5452	4452	4	40	2.25	2
05/31/12	to	06/29/12	07/02/12	to	07/06/12	2138	2098	2	0	0	7590	6550	1	50	3.25	1.25
06/07/12	to	07/06/12	07/09/12	to	07/13/12	(263)	(303)	2	-288	-288	7327	6247	1	50	1.5	1.25
06/14/12	to	07/13/12	07/16/12	to	07/20/12							6247	3	60	2.25	2
06/21/12	to	07/20/12	07/23/12	to	07/27/12	(513)	(573)	3	-1025	-1163	6814	5674	4	60	2	1.75
06/28/12	to	07/27/12	07/30/12	to	08/03/12	1100	1040	3	-238	-238	7914	6714	3	70	2	2
07/05/12	to	08/03/12	08/06/12	to	08/10/12							6714	1	60	2.25	3.5
07/12/12	to	08/10/12	08/13/12	to	08/17/12							6714	1	60	2.25	3.5
07/19/12	to	08/17/12	08/20/12	to	08/24/12	(513)	(553)	2	-575	-575	7401	6161	3	40	1.75	1.75
07/26/12	to	08/24/12	08/27/12	to	08/31/12							6161	1	60	2.25	3.5
08/02/12	to	08/31/12	09/03/12	to	09/07/12							6161	3	20	3.25	2.75
08/09/12	to	09/07/12	09/10/12	to	09/14/12	413	393	1	0	0	7814	6554	4	50	2	2
08/16/12	to	09/14/12	09/17/12	to	09/21/12							6554	4	50	2.5	2
08/23/12	to	09/21/12	09/24/12	to	09/28/12							6554	4	50	2.5	2.75
08/30/12	to	09/28/12	10/01/12	to	10/05/12							6554	4	50	2.5	2.75
09/06/12	to	10/05/12	10/08/12	to	10/12/12							6554	3	20	2.5	2.25
09/13/12	to	10/12/12	10/15/12	to	10/19/12							6554	3	20	2	2.25
09/20/12	to	10/19/12	10/22/12	to	10/26/12	(163)	(183)	1	-163	-163	7651	6371	1	30	1.75	1
09/27/12	to	10/26/12	10/29/12	to	11/02/12	288	268	1	0	0	7939	6639	4	30	1.75	1.25
10/04/12	to	11/02/12	11/05/12	to	11/09/12							6639	2	30	1.5	1.25
10/11/12	to	11/09/12	11/12/12	to	11/16/12	(288)	(328)	2	-325	-325	7651	6311	4	50	1.5	1.25
10/18/12	to	11/16/12	11/19/12	to	11/23/12	(138)	(178)	2	-88	-138	7513	6133	4	20	1.75	1.25
10/25/12	to	11/23/12	11/26/12	to	11/30/12							6133	2	70	1.25	1.25
11/01/12	to	11/30/12	12/03/12	to	12/07/12	1150	1110	2	0	0	8663	7243	4	70	1.5	1.5
11/08/12	to	12/07/12	12/10/12	to	12/14/12	(25)	(65)	2	-250	-250	8638	7178	1	30	1.25	1
11/15/12	to	12/14/12	12/17/12	to	12/21/12							7178	2	20	1.5	1.75
11/22/12	to	12/21/12	12/24/12	to	12/28/12	50	30	1	0	0	8688	7208	2	20	1.5	1.75
11/29/12	to	12/28/12	12/31/12	to	01/04/13	113	53	3	-363	-450	8801	7261	2	20	1.5	2
12/06/12	to	01/04/13	01/07/13	to	01/11/13	1638	1598	2	0	0	10439	8859	4	40	1.5	1.75
12/13/12	to	01/11/13	01/14/13	to	01/18/13	(550)	(590)	2	-450	-550	9889	8269	2	60	2.75	1.75

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
12/20/12	to	01/18/13	01/21/13	to	01/25/13	(550)	(570)	1	-550	-550	9339	7699	2	70	2.5	2
12/27/12	to	01/25/13	01/28/13	to	02/01/13							7699	1	20	2.5	3
01/03/13	to	02/01/13	02/04/13	to	02/08/13	1488	1448	2	0	0	10827	9147	1	20	1.5	2.75
01/10/13	to	02/08/13	02/11/13	to	02/15/13	(700)	(740)	2	-488	-700	10127	8407	4	20	1.75	1.75
01/17/13	to	02/15/13	02/18/13	to	02/22/13							8407	2	60	2	2.5
01/24/13	to	02/22/13	02/25/13	to	03/01/13	1050	1010	2	0	0	11177	9417	2	60	2.25	1.75
01/31/13	to	03/01/13	03/04/13	to	03/08/13	(163)	(203)	2	-313	-313	11014	9214	4	30	2.25	2.25
02/07/13	to	03/08/13	03/11/13	to	03/15/13							9214	4	20	2.75	3
02/14/13	to	03/15/13	03/18/13	to	03/22/13							9214	4	20	2.75	3
02/21/13	to	03/22/13	03/25/13	to	03/29/13							9214	4	70	2.25	2.5
02/28/13	to	03/29/13	04/01/13	to	04/05/13	2113	2033	4	-25	-25	13127	11247	4	20	2.75	2.25
03/07/13	to	04/05/13	04/08/13	to	04/12/13							11247	1	50	2	1
03/14/13	to	04/12/13	04/15/13	to	04/19/13	463	443	1	0	0	13590	11690	1	50	2	1.75
03/21/13	to	04/19/13	04/22/13	to	04/26/13							11690	1	50	1.75	1.5
03/28/13	to	04/26/13	04/29/13	to	05/03/13	(338)	(418)	4	-875	-875	13252	11272	1	50	1.75	1.5
04/04/13	to	05/03/13	05/06/13	to	05/10/13	588	528	3	-413	-413	13840	11800	4	70	2	1.5
04/11/13	to	05/10/13	05/13/13	to	05/17/13	(488)	(508)	1	-488	-488	13352	11292	4	20	2	2
04/18/13	to	05/17/13	05/20/13	to	05/24/13	538	518	1	0	0	13890	11810	1	30	1.75	3.25
04/25/13	to	05/24/13	05/27/13	to	05/31/13	(113)	(133)	1	-113	-113	13777	11677	1	30	1.75	3.5
05/02/13	to	05/31/13	06/03/13	to	06/07/13	350	290	3	-288	-313	14127	11967	2	70	2	1.75
05/09/13	to	06/07/13	06/10/13	to	06/14/13	325	305	1	0	0	14452	12272	1	40	1.5	2
05/16/13	to	06/14/13	06/17/13	to	06/21/13	250	170	4	-363	-363	14702	12442	1	60	1	2.5
05/23/13	to	06/21/13	06/24/13	to	06/28/13	(50)	(90)	2	-63	-63	14652	12352	1	70	2	1.25
05/30/13	to	06/28/13	07/01/13	to	07/05/13	1163	1143	1	0	0	15815	13495	1	30	2	2.5
06/06/13	to	07/05/13	07/08/13	to	07/12/13	163	143	1	0	0	15978	13638	4	40	2.75	2.5
06/13/13	to	07/12/13	07/15/13	to	07/19/13	(388)	(408)	1	-388	-388	15590	13230	2	30	2.25	2.5
06/20/13	to	07/19/13	07/22/13	to	07/26/13							13230	4	50	2.25	3
06/27/13	to	07/26/13	07/29/13	to	08/02/13	200	160	2	-38	-38	15790	13390	4	50	2.25	3
07/04/13	to	08/02/13	08/05/13	to	08/09/13							13390	3	50	2.25	2.5
07/11/13	to	08/09/13	08/12/13	to	08/16/13	(363)	(403)	2	-938	-938	15427	12987	1	60	1.25	1.5
07/18/13	to	08/16/13	08/19/13	to	08/23/13							12987	1	50	1.5	2.5
07/25/13	to	08/23/13	08/26/13	to	08/30/13	0	(20)	1	0	0	15427	12967	1	50	1.5	2.5
08/01/13	to	08/30/13	09/02/13	to	09/06/13	138	118	1	0	0	15565	13085	4	20	2.25	3.5
08/08/13	to	09/06/13	09/09/13	to	09/13/13							13085	4	20	2	3
08/15/13	to	09/13/13	09/16/13	to	09/20/13	763	743	1	0	0	16328	13828	4	20	2	3
08/22/13	to	09/20/13	09/23/13	to	09/27/13							13828	4	30	2.5	2.75
08/29/13	to	09/27/13	09/30/13	to	10/04/13							13828	4	30	2.5	2.75
09/05/13	to	10/04/13	10/07/13	to	10/11/13							13828	3	30	2	3
09/12/13	to	10/11/13	10/14/13	to	10/18/13							13828	3	30	1.75	3
09/19/13	to	10/18/13	10/21/13	to	10/25/13	1250	1150	5	0	0	17578	14978	4	20	1.25	1.5
09/26/13	to	10/25/13	10/28/13	to	11/01/13	(600)	(660)	3	-388	-600	16978	14318	4	20	2	2
10/03/13	to	11/01/13	11/04/13	to	11/08/13	1988	1908	4	0	0	18966	16226	4	40	1.75	2.75
10/10/13	to	11/08/13	11/11/13	to	11/15/13							16226	4	70	3	3
10/17/13	to	11/15/13	11/18/13	to	11/22/13	688	668	1	0	0	19654	16894	4	70	3	3
10/24/13	to	11/22/13	11/25/13	to	11/29/13							16894	4	70	3	3.5
10/31/13	to	11/29/13	12/02/13	to	12/06/13	863	843	1	0	0	20517	17737	4	70	3	3.5
11/07/13	to	12/06/13	12/09/13	to	12/13/13							17737	4	70	2.75	3.5
11/14/13	to	12/13/13	12/16/13	to	12/20/13	625	585	2	-113	-113	21142	18322	2	40	2	1.5

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
11/21/13	to	12/20/13	12/23/13	to	12/27/13							18322	4	20	3.25	2.75
11/28/13	to	12/27/13	12/30/13	to	01/03/14							18322	4	20	3.25	2.75
12/05/13	to	01/03/14	01/06/14	to	01/10/14	(475)	(515)	2	-338	-475	20667	17807	3	20	3.5	3.5
12/12/13	to	01/10/14	01/13/14	to	01/17/14							17807	3	20	2.5	2
12/19/13	to	01/17/14	01/20/14	to	01/24/14							17807	3	20	2.5	2
12/26/13	to	01/24/14	01/27/14	to	01/31/14							17807	3	20	2.5	2
01/02/14	to	01/31/14	02/03/14	to	02/07/14	(1825)	(1925)	5	-763	-1825	18842	15882	3	20	2.5	2
01/09/14	to	02/07/14	02/10/14	to	02/14/14							15882	4	40	2	2.75
01/16/14	to	02/14/14	02/17/14	to	02/21/14							15882	3	20	2	2.5
01/23/14	to	02/21/14	02/24/14	to	02/28/14							15882	3	20	2	2.5
01/30/14	to	02/28/14	03/03/14	to	03/07/14	688	668	1	0	0	19530	16550	3	20	2	2.5
02/06/14	to	03/07/14	03/10/14	to	03/14/14	(13)	(53)	2	-25	-25	19517	16497	3	20	2	2.5
02/13/14	to	03/14/14	03/17/14	to	03/21/14	325	305	1	0	0	19842	16802	1	50	2.25	1.5
02/20/14	to	03/21/14	03/24/14	to	03/28/14							16802	4	70	2	1.75
02/27/14	to	03/28/14	03/31/14	to	04/04/14							16802	4	60	1.75	1.75
03/06/14	to	04/04/14	04/07/14	to	04/11/14	200	180	1	0	0	20042	16982	3	40	1.75	1.5
03/13/14	to	04/11/14	04/14/14	to	04/18/14							16982	2	40	1.5	1.25
03/20/14	to	04/18/14	04/21/14	to	04/25/14							16982	2	40	1.5	1.5
03/27/14	to	04/25/14	04/28/14	to	05/02/14	488	428	3	-125	-125	20530	17410	3	20	1.25	1.75
04/03/14	to	05/02/14	05/05/14	to	05/09/14	188	128	3	-675	-675	20718	17538	3	20	1.5	1.75
04/10/14	to	05/09/14	05/12/14	to	05/16/14							17538	2	40	1.25	2.75
04/17/14	to	05/16/14	05/19/14	to	05/23/14							17538	2	40	1.25	2.75
04/24/14	to	05/23/14	05/26/14	to	05/30/14							17538	2	40	1.25	2.75
05/01/14	to	05/30/14	06/02/14	to	06/06/14	925	885	2	-438	-438	21643	18423	2	40	1.25	2.75
05/08/14	to	06/06/14	06/09/14	to	06/13/14							18423	2	20	2	2.5
05/15/14	to	06/13/14	06/16/14	to	06/20/14							18423	4	30	1.75	2.25
05/22/14	to	06/20/14	06/23/14	to	06/27/14							18423	4	30	1.75	2.25
05/29/14	to	06/27/14	06/30/14	to	07/04/14							18423	4	30	1.75	2.25
06/05/14	to	07/04/14	07/07/14	to	07/11/14							18423	4	30	1.75	2.25
06/12/14	to	07/11/14	07/14/14	to	07/18/14							18423	3	70	1	1
06/19/14	to	07/18/14	07/21/14	to	07/25/14							18423	3	70	0.75	1
06/26/14	to	07/25/14	07/28/14	to	08/01/14	(38)	(58)	1	-38	-38	21605	18365	2	50	0.75	1
07/03/14	to	08/01/14	08/04/14	to	08/08/14							18365	3	30	1.25	2
07/10/14	to	08/08/14	08/11/14	to	08/15/14	(350)	(390)	2	-350	-350	21255	17975	3	30	1.25	2
07/17/14	to	08/15/14	08/18/14	to	08/22/14							17975	4	40	1.75	2
07/24/14	to	08/22/14	08/25/14	to	08/29/14							17975	4	40	1.75	2
07/31/14	to	08/29/14	09/01/14	to	09/05/14	(1963)	(2063)	5	-938	-1963	19292	15912	4	40	1.75	2
08/07/14	to	09/05/14	09/08/14	to	09/12/14	(100)	(120)	1	-100	-100	19192	15792	3	30	2	1.25
08/14/14	to	09/12/14	09/15/14	to	09/19/14	363	323	2	0	0	19555	16115	3	30	2	1.5
08/21/14	to	09/19/14	09/22/14	to	09/26/14							16115	4	60	2.25	2
08/28/14	to	09/26/14	09/29/14	to	10/03/14	475	455	1	0	0	20030	16570	4	60	2.25	2
09/04/14	to	10/03/14	10/06/14	to	10/10/14	363	343	1	0	0	20393	16913	4	60	2.25	2
09/11/14	to	10/10/14	10/13/14	to	10/17/14	2475	2415	3	0	0	22868	19328	4	60	2.25	2.25
09/18/14	to	10/17/14	10/20/14	to	10/24/14							19328	3	30	2	1.75
09/25/14	to	10/24/14	10/27/14	to	10/31/14	913	853	3	0	0	23781	20181	3	30	2	1.5
10/02/14	to	10/31/14	11/03/14	to	11/07/14	650	570	4	-600	-838	24431	20751	4	40	2	2.25
10/09/14	to	11/07/14	11/10/14	to	11/14/14							20751	3	30	2.5	2.5
10/16/14	to	11/14/14	11/17/14	to	11/21/14							20751	4	20	2.75	3.5

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
10/23/14	to	11/21/14	11/24/14	to	11/28/14							20751	4	20	2.75	3.5
10/30/14	to	11/28/14	12/01/14	to	12/05/14	425	385	2	0	0	24856	21136	4	20	2.75	3.25
11/06/14	to	12/05/14	12/08/14	to	12/12/14							21136	4	30	3.5	2.25
11/13/14	to	12/12/14	12/15/14	to	12/19/14	950	930	1	0	0	25806	22066	4	50	3.5	2.5
11/20/14	to	12/19/14	12/22/14	to	12/26/14							22066	4	50	3.5	2.5
11/27/14	to	12/26/14	12/29/14	to	01/02/15							22066	4	50	3.5	2.5
12/04/14	to	01/02/15	01/05/15	to	01/09/15							22066	4	50	3.5	2.5
12/11/14	to	01/09/15	01/12/15	to	01/16/15							22066	3	70	3.5	3.5
12/18/14	to	01/16/15	01/19/15	to	01/23/15	788	648	7	-800	-1438	26594	22714	3	20	2.25	2.25
12/25/14	to	01/23/15	01/26/15	to	01/30/15							22714	4	50	3.5	2.75
01/01/15	to	01/30/15	02/02/15	to	02/06/15	350	290	3	-250	-350	26944	23004	4	50	3.5	2.75
01/08/15	to	02/06/15	02/09/15	to	02/13/15							23004	4	70	3.25	2.25
01/15/15	to	02/13/15	02/16/15	to	02/20/15							23004	4	70	3.25	2.25
01/22/15	to	02/20/15	02/23/15	to	02/27/15							23004	3	60	3	2.25
01/29/15	to	02/27/15	03/02/15	to	03/06/15	800	760	2	0	0	27744	23764	4	50	3	2.75
02/05/15	to	03/06/15	03/09/15	to	03/13/15	163	143	1	0	0	27907	23907	4	50	3	3
02/12/15	to	03/13/15	03/16/15	to	03/20/15	(363)	(403)	2	-1075	-1075	27544	23504	3	30	3.25	2.75
02/19/15	to	03/20/15	03/23/15	to	03/27/15	(363)	(423)	3	-713	-1063	27181	23081	2	20	3	3
02/26/15	to	03/27/15	03/30/15	to	04/03/15	(13)	(33)	1	-13	-13	27168	23048	2	30	3.25	3
03/05/15	to	04/03/15	04/06/15	to	04/10/15	(588)	(608)	1	-588	-588	26580	22440	1	20	3	2.5
03/12/15	to	04/10/15	04/13/15	to	04/17/15							22440	3	70	3.5	2.25
03/19/15	to	04/17/15	04/20/15	to	04/24/15							22440	3	30	3.5	2.5
03/26/15	to	04/24/15	04/27/15	to	05/01/15	238	138	5	-288	-288	26818	22578	4	30	2.5	3
04/02/15	to	05/01/15	05/04/15	to	05/08/15	1075	1015	3	0	0	27893	23593	4	30	3.5	3
04/09/15	to	05/08/15	05/11/15	to	05/15/15							23593	3	40	3.5	3.25
04/16/15	to	05/15/15	05/18/15	to	05/22/15	1263	1243	1	0	0	29156	24836	3	40	3	3.25
04/23/15	to	05/22/15	05/25/15	to	05/29/15							24836	3	40	3	3.25
04/30/15	to	05/29/15	06/01/15	to	06/05/15	350	310	2	0	0	29506	25146	3	40	3.5	2.75
05/07/15	to	06/05/15	06/08/15	to	06/12/15							25146	3	40	3.5	2.75
05/14/15	to	06/12/15	06/15/15	to	06/19/15	(725)	(745)	1	-725	-725	28781	24401	4	40	1.75	3.5
05/21/15	to	06/19/15	06/22/15	to	06/26/15							24401	2	70	1.5	3.25
05/28/15	to	06/26/15	06/29/15	to	07/03/15	(475)	(515)	2	-275	-475	28306	23886	4	30	2.75	3.5
06/04/15	to	07/03/15	07/06/15	to	07/10/15	(338)	(358)	1	-338	-338	27968	23528	3	20	2	2.5
06/11/15	to	07/10/15	07/13/15	to	07/17/15							23528	1	70	1.5	1.5
06/18/15	to	07/17/15	07/20/15	to	07/24/15							23528	3	20	3.25	2.25
06/25/15	to	07/24/15	07/27/15	to	07/31/15	(125)	(165)	2	-675	-675	27843	23363	3	20	3.25	2
07/02/15	to	07/31/15	08/03/15	to	08/07/15	(950)	(970)	1	-950	-950	26893	22393	4	60	3	2.75
07/09/15	to	08/07/15	08/10/15	to	08/14/15							22393	2	20	2.75	2.25
07/16/15	to	08/14/15	08/17/15	to	08/21/15	413	393	1	0	0	27306	22786	2	20	2.75	2.25
07/23/15	to	08/21/15	08/24/15	to	08/28/15	538	498	2	-125	-125	27844	23284	2	20	2.75	2.25
07/30/15	to	08/28/15	08/31/15	to	09/04/15	(275)	(395)	6	-550	-1138	27569	22889	3	50	1.75	1.75
08/06/15	to	09/04/15	09/07/15	to	09/11/15							22889	2	60	2.75	2.5
08/13/15	to	09/11/15	09/14/15	to	09/18/15	350	330	1	0	0	27919	23219	2	50	2.75	1.75
08/20/15	to	09/18/15	09/21/15	to	09/25/15							23219	2	50	3	1.75
08/27/15	to	09/25/15	09/28/15	to	10/02/15	1925	1885	2	0	0	29844	25104	3	40	1.75	2.25
09/03/15	to	10/02/15	10/05/15	to	10/09/15							25104	2	40	3.5	1.75
09/10/15	to	10/09/15	10/12/15	to	10/16/15	313	293	1	0	0	30157	25397	4	20	2.5	2.5
09/17/15	to	10/16/15	10/19/15	to	10/23/15	1675	1615	3	-63	-63	31832	27012	4	20	2.5	2.5

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
09/24/15	to	10/23/15	10/26/15	to	10/30/15	838	818	1	0	0	32670	27830	2	60	1.5	1.75
10/01/15	to	10/30/15	11/02/15	to	11/06/15	(513)	(573)	3	-263	-513	32157	27257	4	30	3.5	2.75
10/08/15	to	11/06/15	11/09/15	to	11/13/15	(475)	(495)	1	-475	-475	31682	26762	4	30	3	2.25
10/15/15	to	11/13/15	11/16/15	to	11/20/15							26762	4	30	3	2.5
10/22/15	to	11/20/15	11/23/15	to	11/27/15							26762	4	30	3	2.5
10/29/15	to	11/27/15	11/30/15	to	12/04/15	938	758	9	-1400	-1400	32620	27520	3	70	1.25	1.75
11/05/15	to	12/04/15	12/07/15	to	12/11/15							27520	4	40	2.25	3.25
11/12/15	to	12/11/15	12/14/15	to	12/18/15	(963)	(1003)	2	-650	-963	31657	26517	4	70	3.5	3
11/19/15	to	12/18/15	12/21/15	to	12/25/15							26517	3	70	3	3.5
11/26/15	to	12/25/15	12/28/15	to	01/01/16							26517	3	70	3	3.5
12/03/15	to	01/01/16	01/04/16	to	01/08/16							26517	3	70	3	3.5
12/10/15	to	01/08/16	01/11/16	to	01/15/16							26517	4	20	2	2.25
12/17/15	to	01/15/16	01/18/16	to	01/22/16	1625	1585	2	0	0	33282	28102	2	30	1.5	2.25
12/24/15	to	01/22/16	01/25/16	to	01/29/16	431	411	1	0	0	33713	28513	2	50	2	1.75
12/31/15	to	01/29/16	02/01/16	to	02/05/16	1863	1823	2	0	0	35576	30336	2	30	1.5	2
01/07/16	to	02/05/16	02/08/16	to	02/12/16	(213)	(273)	3	-413	-413	35363	30063	1	20	1.75	3
01/14/16	to	02/12/16	02/15/16	to	02/19/16							30063	1	20	2	3
01/21/16	to	02/19/16	02/22/16	to	02/26/16	(369)	(389)	1	-369	-369	34994	29674	1	70	1.25	2.75
01/28/16	to	02/26/16	02/29/16	to	03/04/16	1119	1039	4	-75	-75	36113	30713	4	50	2	2.5
02/04/16	to	03/04/16	03/07/16	to	03/11/16	1556	1416	7	-556	-556	37669	32129	4	60	2.5	3
02/11/16	to	03/11/16	03/14/16	to	03/18/16	863	843	1	0	0	38532	32972	2	40	3	2.75
02/18/16	to	03/18/16	03/21/16	to	03/25/16							32972	2	50	1.75	2.25
02/25/16	to	03/25/16	03/28/16	to	04/01/16	631	611	1	0	0	39163	33583	2	50	1.75	2.25
03/03/16	to	04/01/16	04/04/16	to	04/08/16							33583	2	40	3	2.75
03/10/16	to	04/08/16	04/11/16	to	04/15/16							33583	4	20	3.25	3.5
03/17/16	to	04/15/16	04/18/16	to	04/22/16	1000	980	1	0	0	40163	34563	3	20	3.5	2.25
03/24/16	to	04/22/16	04/25/16	to	04/29/16	(388)	(408)	1	-388	-388	39775	34155	3	20	3.5	2
03/31/16	to	04/29/16	05/02/16	to	05/06/16	(694)	(754)	3	-281	-694	39081	33401	2	40	1.5	1.5
04/07/16	to	05/06/16	05/09/16	to	05/13/16							33401	4	40	3	2
04/14/16	to	05/13/16	05/16/16	to	05/20/16	413	393	1	0	0	39494	33794	4	40	3	2
04/21/16	to	05/20/16	05/23/16	to	05/27/16							33794	3	30	3.25	2
04/28/16	to	05/27/16	05/30/16	to	06/03/16	(831)	(871)	2	-1219	-1219	38663	32923	2	20	2.25	2
05/05/16	to	06/03/16	06/06/16	to	06/10/16							32923	2	20	2.25	3.25
05/12/16	to	06/10/16	06/13/16	to	06/17/16							32923	4	30	3.25	3.25
05/19/16	to	06/17/16	06/20/16	to	06/24/16							32923	4	30	3.25	3.25
05/26/16	to	06/24/16	06/27/16	to	07/01/16	(556)	(616)	3	-394	-556	38107	32307	4	30	3.25	3.25
06/02/16	to	07/01/16	07/04/16	to	07/08/16							32307	3	40	3.25	3.25
06/09/16	to	07/08/16	07/11/16	to	07/15/16							32307	4	20	2.25	2.75
06/16/16	to	07/15/16	07/18/16	to	07/22/16							32307	4	20	2	2.75
06/23/16	to	07/22/16	07/25/16	to	07/29/16							32307	4	20	2	2.75
06/30/16	to	07/29/16	08/01/16	to	08/05/16	(219)	(259)	2	-156	-219	37888	32048	4	50	1.75	3
07/07/16	to	08/05/16	08/08/16	to	08/12/16							32048	3	50	2	1.5
07/14/16	to	08/12/16	08/15/16	to	08/19/16	(381)	(401)	1	-381	-381	37507	31647	3	70	1.75	1.75
07/21/16	to	08/19/16	08/22/16	to	08/26/16	1363	1303	3	-638	-638	38870	32950	4	40	1.5	1
07/28/16	to	08/26/16	08/29/16	to	09/02/16							32950	4	30	3	2.25
08/04/16	to	09/02/16	09/05/16	to	09/09/16							32950	4	30	3	2.25
08/11/16	to	09/09/16	09/12/16	to	09/16/16							32950	4	30	3.25	2.25
08/18/16	to	09/16/16	09/19/16	to	09/23/16							32950	4	30	3.25	2.25

In-Sample Dates			Out-Of-Sample Dates			osnp	NOnp$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
08/25/16	to	09/23/16	09/26/16	to	09/30/16							32950	4	30	3.25	2.25
09/01/16	to	09/30/16	10/03/16	to	10/07/16	44	24	1	0	0	38914	32974	4	50	2.5	2.25
09/08/16	to	10/07/16	10/10/16	to	10/14/16							32974	4	70	2	3.5
09/15/16	to	10/14/16	10/17/16	to	10/21/16	(831)	(891)	3	-806	-831	38083	32083	4	70	1.25	2
09/22/16	to	10/21/16	10/24/16	to	10/28/16							32083	4	30	2	2.5
09/29/16	to	10/28/16	10/31/16	to	11/04/16							32083	4	30	2	2.5
10/06/16	to	11/04/16	11/07/16	to	11/11/16	850	830	1	0	0	38933	32913	3	30	2	1.5
10/13/16	to	11/11/16	11/14/16	to	11/18/16	1400	1340	3	-25	-25	40333	34253	4	40	2	1.25
10/20/16	to	11/18/16	11/21/16	to	11/25/16							34253	3	30	1.75	1.5
10/27/16	to	11/25/16	11/28/16	to	12/02/16	(181)	(221)	2	-438	-438	40152	34032	2	40	1.25	1.25
11/03/16	to	12/02/16	12/05/16	to	12/09/16	1088	988	5	-231	-231	41240	35020	3	30	1.5	1.5
11/10/16	to	12/09/16	12/12/16	to	12/16/16	744	724	1	0	0	41984	35744	3	20	2.5	3
11/17/16	to	12/16/16	12/19/16	to	12/23/16							35744	4	50	3	1.5
11/24/16	to	12/23/16	12/26/16	to	12/30/16	250	230	1	0	0	42234	35974	4	50	3	1.5
12/01/16	to	12/30/16	01/02/17	to	01/06/17	(63)	(103)	2	-350	-350	42171	35871	4	50	3	1.5
12/08/16	to	01/06/17	01/09/17	to	01/13/17	(925)	(945)	1	-925	-925	41246	34926	3	30	3.25	2.25
12/15/16	to	01/13/17	01/16/17	to	01/20/17	175	135	2	-194	-194	41421	35061	3	40	1.75	2.25
12/22/16	to	01/20/17	01/23/17	to	01/27/17							35061	4	60	1.75	2.5
12/29/16	to	01/27/17	01/30/17	to	02/03/17	(100)	(120)	1	-100	-100	41321	34941	4	60	1.75	2.5
01/05/17	to	02/03/17	02/06/17	to	02/10/17							34941	4	60	2.25	2.5
01/12/17	to	02/10/17	02/13/17	to	02/17/17							34941	3	40	2	2.5
01/19/17	to	02/17/17	02/20/17	to	02/24/17	431	411	1	0	0	41752	35352	3	40	2	2.5
01/26/17	to	02/24/17	02/27/17	to	03/03/17							35352	4	40	1.75	2.5
02/02/17	to	03/03/17	03/06/17	to	03/10/17	769	729	2	0	0	42521	36081	3	20	1.75	2
02/09/17	to	03/10/17	03/13/17	to	03/17/17	675	615	3	0	0	43196	36696	3	20	1.5	2
02/16/17	to	03/17/17	03/20/17	to	03/24/17							36696	4	30	2.75	2.75
02/23/17	to	03/24/17	03/27/17	to	03/31/17							36696	4	30	2.75	2.75
03/02/17	to	03/31/17	04/03/17	to	04/07/17							36696	4	30	2.75	2.75
03/09/17	to	04/07/17	04/10/17	to	04/14/17							36696	4	30	2.75	2.75
03/16/17	to	04/14/17	04/17/17	to	04/21/17							36696	4	70	2.5	1.5
03/23/17	to	04/21/17	04/24/17	to	04/28/17	488	428	3	-169	-169	43684	37124	3	30	1.25	1.5
03/30/17	to	04/28/17	05/01/17	to	05/05/17							37124	4	20	1.75	1.75
04/06/17	to	05/05/17	05/08/17	to	05/12/17	256	236	1	0	0	43940	37360	4	20	1.75	1.75
04/13/17	to	05/12/17	05/15/17	to	05/19/17	(444)	(464)	1	-444	-444	43496	36896	1	20	1.75	1
04/20/17	to	05/19/17	05/22/17	to	05/26/17	213	193	1	0	0	43709	37089	2	30	1.75	1.25

Appendix I: *n*th Order Polynomial Next Bar's Forecast Math

What is the nth Order Polynomial ?

The nth Order Polynomial, also called the nth Order Fixed Memory Polynomial, is simply the least square fit of a polynomial of the form $b_0+b_1*t+b_2*t^2+b_3*t^3+\ldots b_n*t^n$ to a ***fixed*** number of past data points. Where **t** is discrete time bars. Time could be daily bars or one minute bars. We use the term "Fixed Memory" to designate that only a fixed number of data points are used to calculate the polynomial coefficients. It is assumed that the time bars occur at fixed intervals of time so tic bars would not be appropriate for this analysis. Least squares is a mathematical technique where the squared vertical distance between the data and the curve that is being fit to the data is minimized. When the net squared distance (also called the sum of the squared errors) is minimized, a unique set of coefficients $b_0,b_1,b_2,\ldots,b_n$ for the polynomial is determined. This type of error minimization is mathematically solvable and is widely used in science and mathematics.

For a 4th order polynomial equation, the least squares coefficients are obtained from the solution of the following matrix equation.

$$\begin{vmatrix} T & \sum t & \sum t^2 & \sum t^3 & \sum t^4 \\ \sum t & \sum t^2 & \sum t^3 & \sum t^4 & \sum t^5 \\ \sum t^2 & \sum t^3 & \sum t^4 & \sum t^5 & \sum t^6 \\ \sum t^3 & \sum t^4 & \sum t^5 & \sum t^6 & \sum t^7 \\ \sum t^4 & \sum t^5 & \sum t^6 & \sum t^7 & \sum t^8 \end{vmatrix} \begin{vmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \\ e_0 \end{vmatrix} = \begin{vmatrix} \sum p(t) \\ \sum (p(t)*t) \\ \sum (p(t)*t^2) \\ \sum (p(t)*t^3) \\ \sum (p(t)*t^4) \end{vmatrix}$$

where

p(T) is the current bar's price, **p(T-1)** is the previous bar's price and **p(1)** is the price **T** bars ago.
T is the number of Bars in the Least Squares estimation
$\sum p(t)$ is the summation of prices from **t=1 to T** bars
$\sum p(t)*t$ is the summation of prices times t from **t=1 to T** bars
$\sum t$ is the summation of the integer **t** from **t=1 to T** bars
$\sum t^2$ is the summation of the integer **t** squared from **t=1 to T** bars
etc.

Once the coefficients to the polynomial have been solved for we generate the forecast for the next bar's price which is given for the equation by:

$$P_f = a_0 + b_0*(T+1) + c_0*(T+1)^2 + d_0*(T+1)^3 + e_0*(T+1)^4$$

Where P_f stands for price forecast.

With these coefficients, we can also generate the forecast for the next bar's ***velocity*** and ***velocity*** by the equations:

$$Velocity(T+1) = dP_f /dt = b_0 + 2c_0*(T+1) + 3d_0*(T+1)^2 + 4e_0*(T+1)^3$$

$$Velocity(t+1) = d^2P_f /d^2t = 2\,c_0 + 6d_0*(T+1) + 12e_0*(T+1)^2$$

We use the next bar forecast because changes in the trend are more quickly reflected in the forecast price, velocity and velocity than in the end point price, velocity and velocity.

Programs that solve for the solution to matrix equations can be found in the book "Numerical Recipes" by W. Press, et. al. However these type of matrix equation solvers are very slow and for these type of problems are unstable. They cause numerical errors and floating point overflows due to matrix inversion ill conditioning which produces false results.

Fortunately these type of problems can be solved by a fast, efficient and accurate algorithm using Discrete

Appendix I: *n*th Order Polynomial Next Bar's Forecast Math

Orthogona1 Legendre Polynomials. This method is explained in detail in Norman Morrison' book entitled "Introduction to Sequential Smoothing and Prediction", Chapter 7 page 223., referenced at the end of this section.

Without going into detail here (see Morrison reference), the polynomial filter can now be represented by:

$$P_e(t) = \sum_{j=0}^{n} \beta_j * \phi_j(t) \quad j=0 \text{ to } n$$

Where **n** is the polynomial order, **T** is the total number of Bars of data used in the Least Squares fit and

$$\beta_j = \sum_{k=0}^{T-1} p(t-T+k)*\phi_j(k)$$

$\phi_j(t)$ = the *normalized* discrete Legendre polynomial. t = an integer from 0 to T

The coefficients, **β_0, β_1, β_2, β_3, , , ,β_n** for a **nth** order polynomial can now be solved for by the equation above and we can generate the forecast for the next bar's close, velocity and velocity which are given by the equations

$$P_F(T+1) = \beta_0*\phi_0(T+1) + \beta_1*\phi_1(T+1) + \beta_2*\phi_2(T+1) + \beta_3*\phi_3(T+1) + \ldots + \beta_n*\phi_n(T+1)$$

$$\text{Velocity} = (dP_F/dt)_{(T+1)} = \beta_1*(d\phi_1/dt)_{(T+1)} + \beta_2*(d\phi_2/dt)_{(T+1)} + \beta_3*(d\phi_3/dt)_{(T+1)} ++ \beta_n*(d\phi_n/dt)_{(T+1)}$$

$$\text{Acceleration} = (d^2P_F/d^2t)_{(T+1)} = \beta_2*(d^2\phi_2/d^2t)_{(T+1)} + \beta_3*(d^2\phi_3/d^2t)_{(T+1)} +\ldots+ \beta_n*(d^2\phi_n/d^2t)_{(T+1)}$$

The nth Order Fixed Memory Forecast Next Bar's Velocity System Defined

The least squares forecast is constructed by solving for the least squares coefficients **β_1, β_2, , , β_n** at each bar using the last **T** bars of closing prices and the Discrete Orthogonal Legendre Polynomial equations for β_j above. Then **Velocity** = **dP$_F$(T+1)/dt** is constructed from the velocity equation above and plotted under the price chart. In general what we will be doing is following the plotted curve of **Velocity** which is calculated at each bar from the previous T bars. When the velocity is greater than a threshold amount ***vup*** we will go long. When the velocity is less than a threshold amount ***-vdn*** we will go short.

Buy Rule:
IF Velocity is greater than the threshold amount ***vup*** then buy at the market.

Sell Rule:
IF Velocity is less than the threshold amount ***-vdn*** then sell at the market.

References

1. Morrison, Norman "Introduction to Sequential Smoothing and Prediction", McGraw-Hill Book Company, New York, 1969.

The *Normalization Multiplier*

What is the Multiplier ?

The n^{th} Order Fixed Memory Polynomial, also called an n^{th} Order Polynomial, is the least square fit of a polynomial of the form $\mathbf{b_0+b_1*t+b_2*t^2 +b_3*t^3+…b_n*t^n}$ to a ***fixed*** number of past data points. Where **t** is discrete time bars. Time could be daily bars or one minute bars. We use the term “Fixed Memory” to designate that only a fixed number of data points are used to calculate the polynomial coefficients. It is assumed that the time bars occur at fixed intervals of time so tic bars would not be appropriate for this analysis. Least squares is a mathematical technique where the squared vertical distance between the data and the curve that is being fit to the data is minimized. When the net squared distance (also called the sum of the squared errors) is minimized, a unique set of coefficients $\mathbf{b_0,b_1,b_2,…,b_n}$ for the polynomial is determined. This type of error minimization is mathematically solvable and is widely used in science and mathematics. Once the $\mathbf{b_n}$ coefficients are found using a lookback period of **T** bars to calculate the $\mathbf{b_n}$ coefficients, then the next bar’s estimate **(T+1)** of the n^{th} order polynomial velocity and acceleration can be easily found by the equations below.

$$\mathbf{Velocity(T+1) = dP_f /dt = b_0 + 2c_0*(T+1) + 3d_0*(T+1)^2 + 4e_0*(T+1)^3 +….+n*b_n*(T+1)^{n-1}}$$

Please see the $\boldsymbol{n^{th}}$ **Order Fixed Memory Polynomial Next Bar’s Forecast Math** section for a more detailed explanation.

For any tradable, the inputs to the polynomial are the **polynomial degree(Order)** and the number or lookback bars **N** (denoted by T in equations above). When we plot the velocity or acceleration we notice that the amplitude, and the maximum and minimum values of the velocity or acceleration vary quite significantly with different degree and N inputs.

Below is a table of the standard deviation(Std) of the 56340 calculated Velocity values for different **degree** and **N** inputs. We used 1min bars of the E-Mini from 8/1/2014 to 2/20/2015 to generate this table. As one can see the standard deviation of the velocity for each degree and N vary greatly. For instance for degree=4, N=20 the Std is 6.8 times the Std for degree=1, N=20. This creates problems when trying to determine the correct ranges for vup/vdn during optimization.

@ES.D 1 min bars Date Range 1140801 to 1150220
Total Number of Bars=56340 Norm=0
FixmVn Multiplier= 1/Std to Scale Velocity pw and N Range to One Std

Degree	N	Std	1/Std
1	20	0.1902	5.2565
1	30	0.1540	6.4916
1	40	0.1328	7.5279
1	50	0.1183	8.4502
1	60	0.1077	9.2880
1	70	0.0996	10.0440
avg		0.1338	7.8430
2	20	0.4351	2.2982
2	30	0.3443	2.9046
2	40	0.2936	3.4060
2	50	0.3003	3.8275
2	60	0.2371	4.2180
2	70	0.2173	4.6010
avg		0.2981	3.5425
3	20	0.7854	1.2732
3	30	0.5933	1.6855
3	40	0.4973	2.0111

3	50	0.4347	2.3005
3	60	0.3949	2.5324
3	70	0.3656	2.7352
avg		0.5119	2.0897
4	20	1.2924	0.7738
4	30	0.9279	1.0777
4	40	0.7582	1.3189
4	5	0.6542	1.5285
4	60	0.5804	1.7228
4	70	0.5314	1.8818
avg		0.7908	1.3839

The problem may get worse when we want to find good inputs for other tradables. Other tradables, because of their scales and tick size have much different Velocity ranges then the E-Mini for the same degree and N. Thus the NS search ranges have to be different for each different tradable.

To solve this problem and to have a standard search space for each tradable, I created a **Mult** input for each FixmXVA Velocity strategy and indicator .. If each tradable's Velocity is multiplied by a number such that the standard deviation of that tradable's Velocity is around one then the search space for vup and vdn for each tradable would be 0 to 3.5 SDs and we wouldn't have to change the TS search space every time we wanted to examine a new stock or future. The complicated equations that I use to normalize the ranges to one standard deviation were derived using the software TableCurve 3D, automated surface and equation discovery. .

Below are the results for multiplying the Velocity by TableCurve 3D formula results.

```
@ES.D 1 min bars Date Range 1140801 to 1150220
Total Number of Bars=56340 Norm=1
Trading Times Constraint Start Time=0 EndTime=0
FixmVn Multiplier to Scale Velocity pw and N Range to One Std
pw N
1 20 Std=1.23911 1/Std Mult=0.807028
1 30 Std=1.29301 1/Std Mult=0.773391
1 40 Std=1.33485 1/Std Mult=0.749148
1 50 Std=1.36727 1/Std Mult=0.731383
1 60 Std=1.39422 1/Std Mult=0.717248
1 70 Std=1.41978 1/Std Mult=0.704335
degree=1 1/Std Mult=0.747089

2 20 Std=1.28656 1/Std Mult=0.777264
2 30 Std=1.2948 1/Std Mult=0.77232
2 40 Std=1.30967 1/Std Mult=0.763551
2 50 Std=1.3304 1/Std Mult=0.751652
2 60 Std=1.34514 1/Std Mult=0.743415
2 70 Std=1.35128 1/Std Mult=0.740039
degree=2 1/Std Mult=0.75804

3 20 Std=1.40562 1/Std Mult=0.711432
3 30 Std=1.33297 1/Std Mult=0.750206
3 40 Std=1.31288 1/Std Mult=0.761682
3 50 Std=1.30074 1/Std Mult=0.768795
3 60 Std=1.30892 1/Std Mult=0.763991
3 70 Std=1.3213 1/Std Mult=0.756829
degree=3 1/Std Mult=0.752156

4 20 Std=1.57493 1/Std Mult=0.634949
4 30 Std=1.40116 1/Std Mult=0.713696
4 40 Std=1.33307 1/Std Mult=0.750151
4 50 Std=1.29424 1/Std Mult=0.772655
4 60 Std=1.26449 1/Std Mult=0.790835
4 70 Std=1.256 1/Std Mult=0.796177
degree=4 1/Std Mult=0.743077

1/Std Mult Ave=0.750091
```

Added Note:
The complicated equations that I use to normalize the ranges to one standard deviation were derived using the software TableCurve 3D, automated surface and equation discovery. These equations are ***independen***t of the tradable and only depend on the polynomial degree and N. If you do not wish to use these equations then set inorm=0. With iNorm=0 you will have to determine the vup/vdn ranges for your optimization. They will be different for different tradables and time bars