

Trading 1Min Bar Euro Futures Using The Nth order Fixed Memory Polynomial Velocity Strategy Part 2

August 1, 2011 –September 29, 2017

Working Paper October 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 my working paper at <http://meyersanalytics.com/publications2/EC1FixmVnXOXT2.pdf> 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 Euro futures intraday. In that working paper, I assumed that strategy inputs that generated high profit factors (**PF**) and high equity correlation coefficients (r^2) in the in-sample section would generate poor profit performance in the out-of-sample-section. I made these *assumptions* because I believed that strategy performance results that curve fitted the price movements and price noise in the in-sample section would produce high **PF**s and high r^2 s metrics and since the same random price noise in the in-sample section would not be repeated, good performance would not be repeated in the out-of-sample section.

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 metrics will be from those strategy input combinations that are able to produce profits from the price pattern *and* the random spurious price movements. While the price patterns will repeat, the same spurious price movements will not. If the spurious price movements that were captured by a certain set of strategy input parameters were a large part of the total net profits, which they are in most price series, then choosing these input parameters will produce losses when traded on future data. These losses occur because the spurious price movements will not be repeated in the same way

In this paper, I tested those previous assumptions by scanning for strategy inputs that generated in-sample performance metrics with high **PF**s and r^2 s and examined how these metric combinations performed in the out-of-sample section. I was surprised, at least for the EC 1-min data examined here, high **PF** and r^2 metric filters in the in-sample section performed much better on out-of-sample data than my previous assumption of eliminating strategy inputs that generated high **PF**s and r^2 s in the in-sample section. Performance results are compared for these different **PF** and r^2 metric filter assumptions.

The n^{th} Order Fixed Memory Strategy Defined

The least squares forecast n^{th} order fixed memory polynomial velocity is constructed by solving for the coefficients $\beta_0, \beta_1, \beta_2, \beta_3 \dots \beta_n$ 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 2nd, 3rd and 4th 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 (1st through 4th) 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 n^{th} order polynomial velocity, the number of lookback prices, **N**, needed to compute the polynomial coefficients and the threshold amounts **vup** and **vdn**. When the n^{th} 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 **7:00am**. 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 Strategy 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 strategy, 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 September 29, 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 September 29, 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 326 weeks of EC (see Walk Forward Testing below) creating 322 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 perform on 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 combinations that are 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 set of strategy inputs, 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://meversanalytics.com/Walk-Forward-Optimization.html> for a listing of these performance metrics).

For our computer run we will have the PWFO breakup the 322 weeks of EC one-minute bar price data into 322 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. (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 322 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 322 in-sample and out-of-sample(oos) sections generated by the PWFO to find the best combination of in-sample section performance metrics, also called a filter, that determines the strategy 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://meversanalytics.com/wfme.html>

For the in-sample data we will run the TradeStation optimization engine on the 322 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 322 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 strategy, one has some combination of performance metrics, 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 strategy and filter. **Mathematical note:** *This assumption assumes that the out-of-sample returns are from probability distributions that have a finite variance.*

Why use the walk forward technique? Why not just perform an optimization on the whole price series and choose the input parameters that give the best total net profits or profit factor? Surely the price noise cancels itself out with such a large number of 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 strategy 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 price movements. While the price patterns will repeat, the same spurious price movements will not. If the spurious price movements that were captured by a certain set of strategy input parameters were a large part of the total net profits, which they are in most price series, then choosing these input parameters will produce losses when traded on future data. These losses occur because the spurious price 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 losses 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.

To gain confidence that our strategy input parameter selection using a combination of performance metrics in 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 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 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 In-Sample 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.

Using the Walk Forward Metric Explorer (WFME) to Find the Best In-Sample Performance Metric Combination

The WFME is described in detail here <http://meyersanalytics.com/wfme.html>.

Here is a combination filter, discovered by the Walk Forward Metric Explorer that is used in this paper with good out-of-sample results. In the in-sample section we eliminate all strategy input rows that have a $PF < 4.5$. That is, we only want strategy rows that have a $PF \geq 4.5$. $R2$ is defined as the in-sample trade equity regression trend line coefficient of correlation r^2 . r^2 is a measure of how well a straight line fits the equity curve generated by a set of in-sample strategy inputs. So, in the in-sample section we eliminate all strategy input rows that have a $R2 < 80$. After using the $PF-R2$ filter, as described, there can still be 100's of rows left in the in-sample section. The PWFO generates the metric $eq10$. This metric is the **Projected Equity 10 Trades in Future Using a 2nd Order Polynomial Line/1000**. Let us choose the 20 rows in the in-sample section that contain the highest(top) value of $eq10$ from the rows that are left after the $PF-R2$ row elimination. This filter will now leave 20 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 of the metric $m(ru-p)$ in the in-sample section. $m(ru-p)$ is the median of all the trades (Maximum Trade Run-up – Final Trade Profit) for a given set of strategy inputs. As we would want the final trade profit to be as close to the maximum runup as possible we look for a minimum value for the metric. This would produce a filter named $t20eq10|pf>4.5| r2>80-m(ru-p)$. 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 filter $t20eq10|pf>4.5| r2>80-m(ru-p)$ finds the strategy inputs parameters in each of the 322 in-sample sections and applies these inputs to each of the 322 out-of-sample sections. Using the filter in-sample strategy inputs on the 322 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 3a 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 86400 different metric filters were examined. We chose **Row 4** 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.

Figure 3b shows a filter computer run with the elimination of strategy inputs that have high **PFs** and high **r²s**, as discussed at the start of this paper, with a small sample of other filter combinations that are constructed in a similar manner. As can be seen by comparing **figures 3a** and **3b** the **high PF**, **high r²** filters gave much better out-of-sample performance.

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 322 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 322 PWFO files. At the end, each of the 5000 mirror filters will have 322 *random* OSNP's picked from the rows of the 322 PWFO files. The sum of the 322 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 322 out-of-sample files of **(\$109.0)** with a bootstrap standard deviation of **\$63.29**. (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/322) and the average net weekly for the filter would be the filter toNP/ (# of OOS) periods traded or $45292/177=255.9$. The probability of obtaining our filters average weekly net profit of **255.9** is 3.89×10^{-9} which is **5.77** standard deviations from the bootstrap average. For our filter, in row 4 in Figure 3, the expected number of cases that we could obtain by pure chance that would match or exceed **\$255.9** is $[1-(1-3.89 \times 10^{-9})^{86400}] \sim 86400 \times 3.89 \times 10^{-9} = 0.00034$ where **86400** 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 322 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 322 weeks ending 8/5/11 to 9/29/17. 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 Strategy 1-minute bar chart of EC from 9/29/2016

Discussion of Strategy Performance

In Figure 3a Row 4 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 28 weeks for this filter. This means that 28 weeks was the longest time that the equity for this strategy failed to make a new equity high in the 322 out-of-sample weeks. For this strategy, the **%P** (% of oos periods that are positive) was 64%, 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 322 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 0.983 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 12 trades/week with an average of 2.3 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 322 out-of-sample weeks the filter only traded 177 of those weeks or 55% of the time with 56% of all oos trades profitable and 64% of all oos 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 \$7100 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 making +\$86 net on 1 trade 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 – 9/29/17**

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.

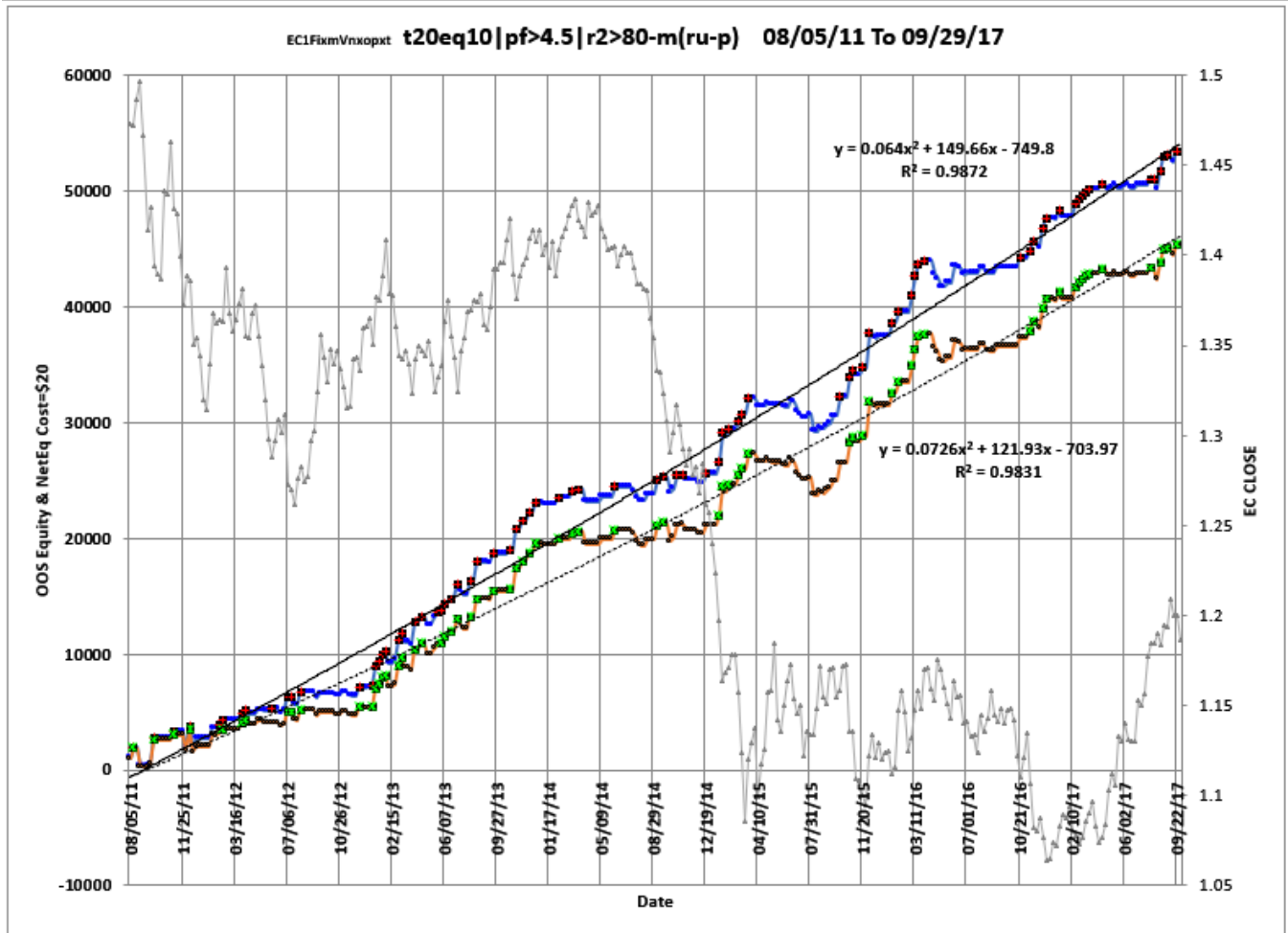
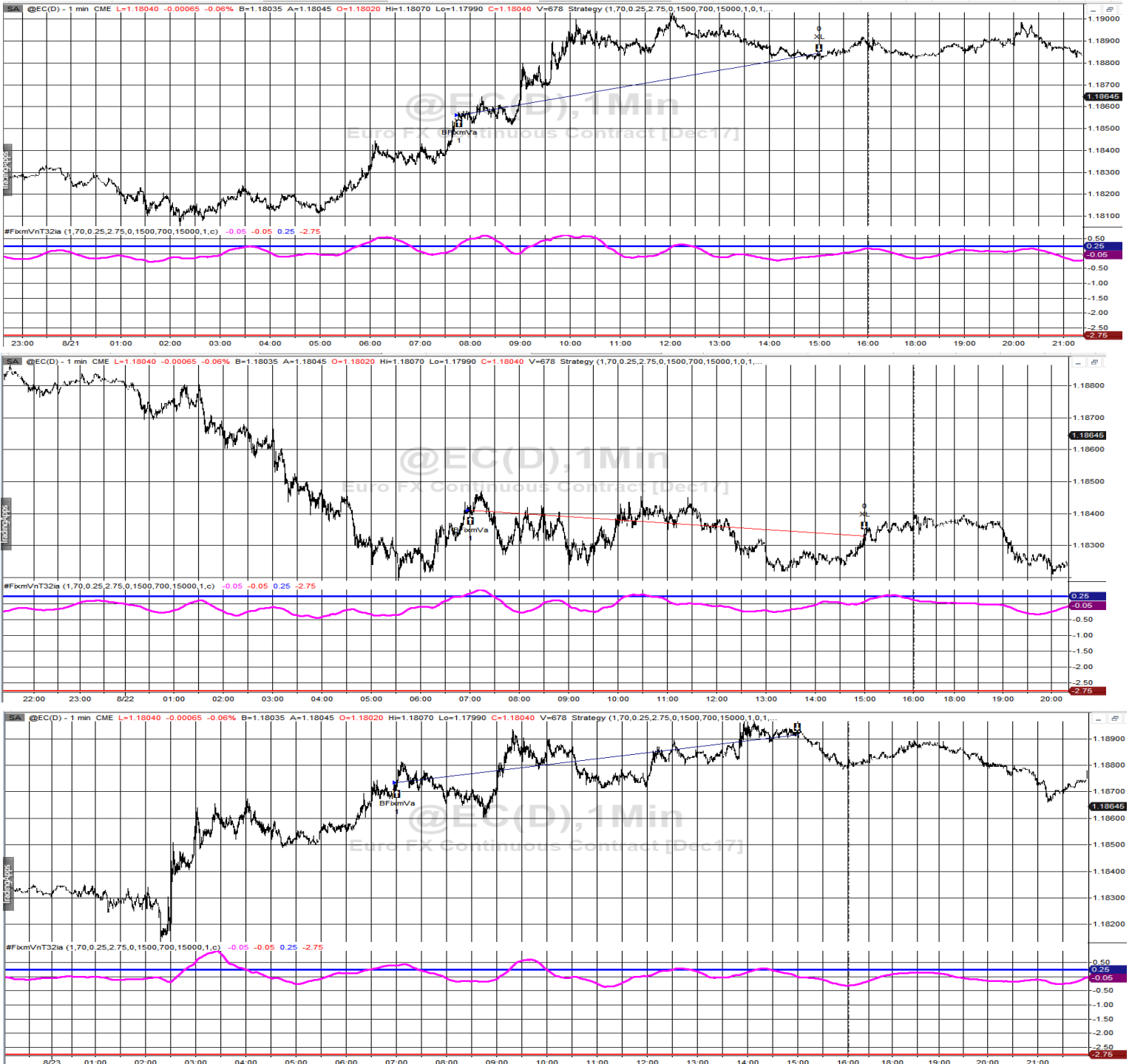
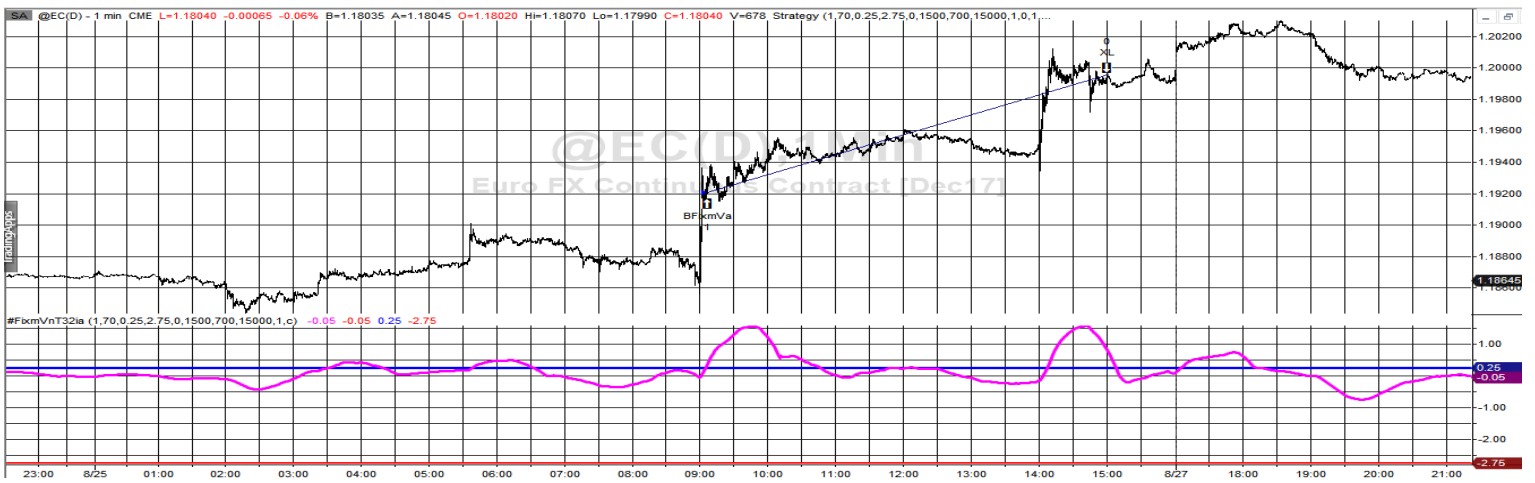
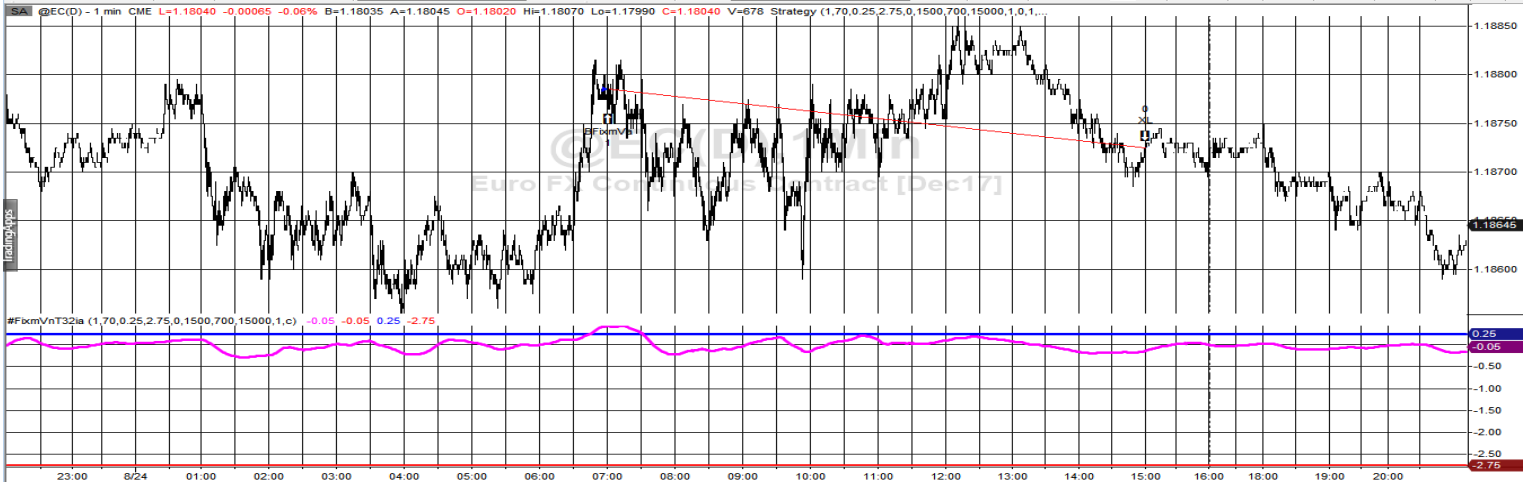


Figure 2 Walk Forward Out-Of-Sample Performance Summary for Nth Order Fixed Memory Polynomial Velocity Strategy EC 1-minute bar chart from 8/21/17 8/25/17 pw=1, n=70, vup=0.25, vdn=-2.75





**Figure 3a Partial output of the Walk Forward Metric Performance Explorer (WFME)
EC1 min bars Nth Order Fixed Memory Velocity Strategy with PF>=, r²>=**

	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	Z	AA
1	EC1FixmVnxopxt	s08/05/11	e09/29/17	#322	AnyTnp									a(109.0)	s63.2	f86400				c=\$20							
2	Filter-Metric	toGP	toNP	aoGP	aoTr	ao#T	std	skew	kur	t	oW oL	%Wtr	%P	LLtr	LLp	eqDD	wpr	lpr	#	%Ab2	eqA2	Dev^2	KTau	eqR2	Blw	BE	Prob
3	t20eq10 pf>4.5 r<5r2>80-m(ru-p)	53392	45292	302	131.8	2.3	766	0.391	3.91	5.24	1.56	56	64	-1575	-1700	-2913	7	4	177	-1.3	0.13	1871	91	99	28	53	3.89E-09
4	t20eq10 pf>4.5 r2>80-m(ru-p)	53392	45292	302	131.8	2.3	766	0.391	3.91	5.24	1.56	56	64	-1575	-1700	-2913	7	4	177	-1.3	0.13	1871	91	99	28	53	3.89E-09
5	t20eq10 pf>4.5 r<3r2>80-m(ru-p)	53267	45247	303	132.8	2.3	768	0.386	3.89	5.23	1.56	56	64	-1575	-1700	-2913	7	4	176	-1.4	0.12	1871	91	99	28	53	3.48E-09
6	t20eq2b1 r<3r2>70-tWb tLb	52882	42382	251	100.7	2.5	733	0.768	4.96	4.97	1.59	52	62	-1275	-1438	-4700	10	7	211	0.9	0.54	2842	84	96	60	57	4.73E-07
7	t20eq2b1 pf>2.5 r<5r2>70-PF	48723	41543	266	135.7	2	696	(0.009)	3.51	5.17	1.43	57	64	-1350	-1625	-2962	8	4	183	-3.1	0.55	2925	88	95	36	54	5.30E-08
8	t20eq2b1 pf>2.5 r2>70-PF	48723	41543	266	135.7	2	696	(0.009)	3.51	5.17	1.43	57	64	-1350	-1625	-2962	8	4	183	-3.1	0.55	2925	88	95	36	54	5.30E-08
9	t20eq2b1 pf>2 r<5r2>70-tWb tLb	52100	41540	247	98.7	2.5	704	0.727	5.14	5.09	1.62	51	63	-1275	-1438	-3963	10	4	211	-0.1	0.64	3225	81	95	65	54	6.51E-07
10	t20eq2b1 pf>2 r2>70-tWb tLb	52100	41540	247	98.7	2.5	704	0.727	5.14	5.09	1.62	51	63	-1275	-1438	-3963	10	4	211	-0.1	0.64	3225	81	95	65	54	6.51E-07

**Figure 3b Partial output of the Walk Forward Metric Performance Explorer (WFME)
EC1 min bars Nth Order Fixed Memory Velocity Strategy with PF<= & r²<=**

	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	Z	AA
1	EC1FixmVnxopxt	s08/05/11	e09/29/17	#322	AnyTnp									a(109.8)	s61.9	f43200				c=\$20							
2	Filter-Metric	toGP	toNP	aoGP	aoTr	ao#T	std	skew	kur	t	oW oL	%Wtr	%P	LLtr	LLp	eqDD	wpr	lpr	#	%Ab2	eqA2	Dev^2	KTau	eqR2	Blw	BE	Prob
3	b20tLb pf<4 r<5r2<90-eq2V	43171	36111	252	122.3	2.1	752	0.189	3.44	4.39	1.43	56	60	-1525	-1963	-2656	12	5	171	-6.8	0.03	1534	87	99	29	74	1.09E-0
4	b20tLb pf<4 r<3r2<90-eq2V	42546	35426	245	119.5	2	750	0.195	3.43	4.3	1.4	56	60	-1525	-1963	-2656	12	5	174	-6.2	-0.02	1456	88	99	29	77	2.09E-0
5	b20m(ru-p) pf<4 r<3r2<90-mWb	56354	34974	206	52.7	3.9	1011	1.038	9.35	3.37	1.29	51	60	-1538	-3575	-6002	8	10	274	-4.4	0.51	5924	64	86	66	120	6.30E-0
6	t20%P pf<4 r<5r2<70-eq2V	46993	34873	198	77.5	2.6	893	1.015	5.65	3.42	1.37	52	54	-1813	-1950	-5540	9	7	237	-7.4	0.53	3762	72	93	66	117	1.67E-0
7	t20%P pf<4 r2<70-eq2V	46993	34873	198	77.5	2.6	893	1.015	5.65	3.42	1.37	52	54	-1813	-1950	-5540	9	7	237	-7.4	0.53	3762	72	93	66	117	1.67E-0
8	t20mKr pf<3 tLb	41404	34844	248	126.2	2	791	1.217	6.42	4.05	1.49	56	56	-1188	-1456	-3750	9	6	167	-11.9	0.05	2298	81	97	40	86	1.36E-0
9	t20mKr pf<3 r<5-tLb	41404	34844	248	126.2	2	791	1.217	6.42	4.05	1.49	56	56	-1188	-1456	-3750	9	6	167	-11.9	0.05	2298	81	97	40	86	1.36E-0
10	t20%P pf<4 r<3r2<70-eq2V	46930	34810	198	77.4	2.6	893	1.014	5.64	3.41	1.37	52	54	-1813	-1950	-5540	9	7	237	-7.4	0.53	3751	72	93	66	117	1.70E-0

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 (9/29/17), **Number of weeks** (#322), **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. Row4 filter **t20eq10|pf>4.5|r2>80-m(ru-p)**

The **t20eq10|pf>4.5|r2>80-m(ru-p)** filter produced the following average 322 week statistics on Row 4.

toGP = Total out-of-sample(oos) gross profit for these 322 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 322 weeks

aoTr = Average oos profit per trade

ao#T = Average number of oos trades per week

std = standard deviation of weekly oos gross profit

skew= Skew of weekly oos gross profit

Kur = Kurtosis of weekly oos gross profit. Kur=3 is a normal distribution. Kur>3 means fatter tails.

t = The student t statistic for the 322 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.

%Ab2 = The % amount the final oos equity is above or below a 2nd order polynomial fit to the equity curve. Note if final equity is way above the polynomial, it may mean the equity may fall back to the polynomial curve thus indicating possible poor near term future performance.

eqA2 = The ending acceleration of 2nd order polynomial that is fit to the oos equity curve

Dev² = A measure of equity curve smoothness. The square root of the average [(equity curve minus a straight line)²]

KTau² = The Kendall rank coefficient is often used as a test statistic in a statistical hypothesis test to establish whether two variables may be regarded as statistically dependent. This test is non-parametric, as it does not rely on any assumptions on the distributions of X or Y or the distribution of (X,Y). Ktau is a non parametric estimate on the smoothness of the equity curve. For instance, if the equity curve is smooth but not a straight line and curves up, the straight line correlation, **r²**, will go down but the **KTau** will be high.

eqR2 = The correlation coefficient(**r²**) of a straight line fit to the oos equity curve

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 8/6/2011 - 9/29/2017. The input values *degree(pw)*, *N*, *vup*, *vdn* are the values found from applying the filter to the in-sample section optimization runs.

Filter= $t20eq10|pf>4.5|r2>80-m(ru-p)$

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 Date	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	1013	973	2	0	0	1013	973	1	40	3	2
07/07/11	to	08/05/11	08/08/11	to	08/12/11	913	893	1	0	0	1926	1866	4	70	3.5	2.75
07/14/11	to	08/12/11	08/15/11	to	08/19/11						1926	1866	1	40	3.25	2.25
07/21/11	to	08/19/11	08/22/11	to	08/26/11	(1575)	(1595)	1	-1575	-1575	351	271	2	60	3.5	2.25
07/28/11	to	08/26/11	08/29/11	to	09/02/11						351	271	4	70	3	3.25
08/04/11	to	09/02/11	09/05/11	to	09/09/11						351	271	3	70	3	3.25
08/11/11	to	09/09/11	09/12/11	to	09/16/11	213	173	2	0	0	564	444	4	60	3	3
08/18/11	to	09/16/11	09/19/11	to	09/23/11	2250	2170	4	0	0	2814	2614	2	20	2.25	2.5
08/25/11	to	09/23/11	09/26/11	to	09/30/11						2814	2614	1	70	3	3
09/01/11	to	09/30/11	10/03/11	to	10/07/11	(13)	(33)	1	-13	-13	2801	2581	1	70	3	3
09/08/11	to	10/07/11	10/10/11	to	10/14/11						2801	2581	1	60	3.25	3.25
09/15/11	to	10/14/11	10/17/11	to	10/21/11						2801	2581	1	60	3.25	3.25
09/22/11	to	10/21/11	10/24/11	to	10/28/11						2801	2581	1	40	2.75	2.75
09/29/11	to	10/28/11	10/31/11	to	11/04/11	475	455	1	0	0	3276	3036	1	50	2.75	2.5
10/06/11	to	11/04/11	11/07/11	to	11/11/11						3276	3036	1	70	2.25	3.25
10/13/11	to	11/11/11	11/14/11	to	11/18/11						3276	3036	1	70	2.25	2.5
10/20/11	to	11/18/11	11/21/11	to	11/25/11						3276	3036	1	70	2.25	2.5
10/27/11	to	11/25/11	11/28/11	to	12/02/11	(1288)	(1368)	4	-563	-1563	1988	1668	3	50	3.25	2.25
11/03/11	to	12/02/11	12/05/11	to	12/09/11	1813	1773	2	0	0	3801	3441	4	20	2.75	2
11/10/11	to	12/09/11	12/12/11	to	12/16/11	(1700)	(1920)	11	-600	-2113	2101	1521	3	30	1.75	1.25
11/17/11	to	12/16/11	12/19/11	to	12/23/11	588	528	3	-25	-25	2689	2049	2	20	2.5	1
11/24/11	to	12/23/11	12/26/11	to	12/30/11						2689	2049	4	20	2.5	1.75
12/01/11	to	12/30/11	01/02/12	to	01/06/12						2689	2049	4	70	2.25	3
12/08/11	to	01/06/12	01/09/12	to	01/13/12						2689	2049	4	30	2.75	2
12/15/11	to	01/13/12	01/16/12	to	01/20/12						2689	2049	3	20	3.5	2
12/22/11	to	01/20/12	01/23/12	to	01/27/12	963	943	1	0	0	3652	2992	4	70	1.75	2.5
12/29/11	to	01/27/12	01/30/12	to	02/03/12	0	(60)	3	-313	-313	3652	2932	3	40	1.75	2
01/05/12	to	02/03/12	02/06/12	to	02/10/12	250	210	2	0	0	3902	3142	3	20	2.25	3.25
01/12/12	to	02/10/12	02/13/12	to	02/17/12	438	358	4	-513	-688	4340	3500	3	70	0.75	2.25
01/19/12	to	02/17/12	02/20/12	to	02/24/12						4340	3500	3	70	2	3
01/26/12	to	02/24/12	02/27/12	to	03/02/12						4340	3500	2	60	1.75	2.75
02/02/12	to	03/02/12	03/05/12	to	03/09/12						4340	3500	2	60	2	2.75
02/09/12	to	03/09/12	03/12/12	to	03/16/12						4340	3500	3	40	2.5	3

In-Sample Date			Out-of-Sample Dates			osnp	NOnp\$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
02/16/12	to	03/16/12	03/19/12	to	03/23/12						4340	3500	3	30	2.5	2.5
02/23/12	to	03/23/12	03/26/12	to	03/30/12	575	555	1	0	0	4915	4055	1	60	1.5	2
03/01/12	to	03/30/12	04/02/12	to	04/06/12	288	268	1	0	0	5203	4323	1	70	1.75	1.75
03/08/12	to	04/06/12	04/09/12	to	04/13/12	(400)	(420)	1	-400	-400	4803	3903	3	20	2.75	2
03/15/12	to	04/13/12	04/16/12	to	04/20/12						4803	3903	3	30	2.5	1.5
03/22/12	to	04/20/12	04/23/12	to	04/27/12						4803	3903	2	60	1.75	1.5
03/29/12	to	04/27/12	04/30/12	to	05/04/12	388	348	2	0	0	5191	4251	4	20	1.5	3.25
04/05/12	to	05/04/12	05/07/12	to	05/11/12						5191	4251	4	20	1.5	3.25
04/12/12	to	05/11/12	05/14/12	to	05/18/12	(50)	(170)	6	-263	-400	5141	4081	4	20	1.75	2.25
04/19/12	to	05/18/12	05/21/12	to	05/25/12						5141	4081	2	20	2.25	2.25
04/26/12	to	05/25/12	05/28/12	to	06/01/12	88	8	4	-163	-275	5229	4089	4	50	2	1.75
05/03/12	to	06/01/12	06/04/12	to	06/08/12						5229	4089	4	20	3.25	2.25
05/10/12	to	06/08/12	06/11/12	to	06/15/12						5229	4089	1	60	1.25	2.25
05/17/12	to	06/15/12	06/18/12	to	06/22/12	(300)	(340)	2	-250	-300	4929	3749	1	30	1.5	2.25
05/24/12	to	06/22/12	06/25/12	to	06/29/12	225	205	1	0	0	5154	3954	1	60	1.5	1.75
05/31/12	to	06/29/12	07/02/12	to	07/06/12	1150	1110	2	0	0	6304	5064	1	70	3	1.25
06/07/12	to	07/06/12	07/09/12	to	07/13/12	25	5	1	0	0	6329	5069	2	70	2.25	1.75
06/14/12	to	07/13/12	07/16/12	to	07/20/12	(675)	(735)	3	-713	-1138	5654	4334	1	60	2.75	1.25
06/21/12	to	07/20/12	07/23/12	to	07/27/12	(50)	(90)	2	-63	-63	5604	4244	3	70	1.75	2.75
06/28/12	to	07/27/12	07/30/12	to	08/03/12	1150	970	9	-1288	-1288	6754	5214	3	40	1.25	1.75
07/05/12	to	08/03/12	08/06/12	to	08/10/12						6754	5214	1	20	2.25	3.25
07/12/12	to	08/10/12	08/13/12	to	08/17/12						6754	5214	2	60	2.75	2.75
07/19/12	to	08/17/12	08/20/12	to	08/24/12						6754	5214	2	60	2.5	2.75
07/26/12	to	08/24/12	08/27/12	to	08/31/12						6754	5214	2	60	2.5	2.75
08/02/12	to	08/31/12	09/03/12	to	09/07/12	(475)	(495)	1	-475	-475	6279	4719	3	70	3	1.25
08/09/12	to	09/07/12	09/10/12	to	09/14/12	288	228	3	-600	-600	6567	4947	1	40	1.5	2
08/16/12	to	09/14/12	09/17/12	to	09/21/12						6567	4947	1	40	1.75	1.75
08/23/12	to	09/21/12	09/24/12	to	09/28/12						6567	4947	2	60	1.75	2
08/30/12	to	09/28/12	10/01/12	to	10/05/12						6567	4947	2	60	1.75	2
09/06/12	to	10/05/12	10/08/12	to	10/12/12						6567	4947	1	50	1.75	1.75
09/13/12	to	10/12/12	10/15/12	to	10/19/12	(125)	(145)	1	-125	-125	6442	4802	2	40	2	1.25
09/20/12	to	10/19/12	10/22/12	to	10/26/12						6442	4802	3	30	1.75	1.5
09/27/12	to	10/26/12	10/29/12	to	11/02/12	300	280	1	0	0	6742	5082	2	30	1.75	1.5
10/04/12	to	11/02/12	11/05/12	to	11/09/12						6742	5082	2	70	1.25	1.25
10/11/12	to	11/09/12	11/12/12	to	11/16/12	(288)	(328)	2	-325	-325	6454	4754	4	50	1.75	1.25
10/18/12	to	11/16/12	11/19/12	to	11/23/12	(38)	(58)	1	-38	-38	6416	4696	4	30	1.25	1.25
10/25/12	to	11/23/12	11/26/12	to	11/30/12						6416		4696			
11/01/12	to	11/30/12	12/03/12	to	12/07/12	750	710	2	-63	-63	7166	5406	2	50	1.25	1.5
11/08/12	to	12/07/12	12/10/12	to	12/14/12	(75)	(95)	1	-75	-75	7091	5311	4	20	1.5	1.5
11/15/12	to	12/14/12	12/17/12	to	12/21/12						7091	5311	3	30	1.5	1.5
11/22/12	to	12/21/12	12/24/12	to	12/28/12	50	30	1	0	0	7141	5341	3	30	1.5	1.5
11/29/12	to	12/28/12	12/31/12	to	01/04/13	150	90	3	-400	-488	7291	5431	3	30	1.25	1.75
12/06/12	to	01/04/13	01/07/13	to	01/11/13	1638	1598	2	0	0	8929	7029	4	40	1.5	2.25
12/13/12	to	01/11/13	01/14/13	to	01/18/13	475	415	3	0	0	9404	7444	4	40	1.75	2
12/20/12	to	01/18/13	01/21/13	to	01/25/13	588	528	3	-113	-113	9992	7972	4	20	1	2.5
12/27/12	to	01/25/13	01/28/13	to	02/01/13	213	173	2	-175	-175	10205	8145	3	20	1.25	3
01/03/13	to	02/01/13	02/04/13	to	02/08/13	(963)	(1003)	2	-1175	-1175	9242	7142	4	40	1.25	3
01/10/13	to	02/08/13	02/11/13	to	02/15/13						9242	7142	2	60	2.75	2.5
01/17/13	to	02/15/13	02/18/13	to	02/22/13	238	218	1	0	0	9480	7360	3	40	2.5	2
01/24/13	to	02/22/13	02/25/13	to	03/01/13	1688	1668	1	0	0	11168	9028	1	40	2	1.75
01/31/13	to	03/01/13	03/04/13	to	03/08/13	613	573	2	0	0	11781	9601	1	40	2	1.75
02/07/13	to	03/08/13	03/11/13	to	03/15/13	(713)	(753)	2	-463	-713	11068	8848	3	20	2.5	2.25
02/14/13	to	03/15/13	03/18/13	to	03/22/13						11068	8848	3	40	2.25	2.5
02/21/13	to	03/22/13	03/25/13	to	03/29/13	(263)	(283)	1	-263	-263	10805	8565	4	60	2.25	2.25

In-Sample Date			Out-of-Sample Dates			osnp	NOnp\$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
02/28/13	to	03/29/13	04/01/13	to	04/05/13	1900	1840	3	-100	-100	12705	10405	4	60	2.25	3
03/07/13	to	04/05/13	04/08/13	to	04/12/13						12705	10405	1	60	2	3.25
03/14/13	to	04/12/13	04/15/13	to	04/19/13	513	493	1	0	0	13218	10898	1	70	1.75	1.5
03/21/13	to	04/19/13	04/22/13	to	04/26/13						13218	10898	3	20	1.75	1.5
03/28/13	to	04/26/13	04/29/13	to	05/03/13	(663)	(903)	12	-913	-1588	12555	9995	3	20	1.5	1.25
04/04/13	to	05/03/13	05/06/13	to	05/10/13						12555	9995	2	60	2.25	3.25
04/11/13	to	05/10/13	05/13/13	to	05/17/13	600	500	5	-150	-150	13155	10495	3	40	2	1
04/18/13	to	05/17/13	05/20/13	to	05/24/13	425	265	8	-650	-725	13580	10760	3	40	1.75	0.75
04/25/13	to	05/24/13	05/27/13	to	05/31/13	188	168	1	0	0	13768	10928	3	50	2	3.25
05/02/13	to	05/31/13	06/03/13	to	06/07/13	625	605	1	0	0	14393	11533	2	70	2.25	3
05/09/13	to	06/07/13	06/10/13	to	06/14/13						14393	11533	2	30	3.25	2
05/16/13	to	06/14/13	06/17/13	to	06/21/13	363	323	2	-563	-563	14756	11856	2	30	3.25	1.75
05/23/13	to	06/21/13	06/24/13	to	06/28/13						14756	11856	3	40	3	2
05/30/13	to	06/28/13	07/01/13	to	07/05/13	1263	1243	1	0	0	16019	13099	3	50	3	2
06/06/13	to	07/05/13	07/08/13	to	07/12/13	(513)	(533)	1	-513	-513	15506	12566	3	20	3	2.75
06/13/13	to	07/12/13	07/15/13	to	07/19/13	(388)	(408)	1	-388	-388	15118	12158	1	20	3	2.25
06/20/13	to	07/19/13	07/22/13	to	07/26/13						15118	12158	4	70	1.5	2
06/27/13	to	07/26/13	07/29/13	to	08/02/13	1163	1063	5	-400	-400	16281	13221	4	70	1.25	2
07/04/13	to	08/02/13	08/05/13	to	08/09/13						16281	13221	4	70	2.5	2.5
07/11/13	to	08/09/13	08/12/13	to	08/16/13	1663	1563	5	-225	-225	17944	14784	2	20	0.5	3
07/18/13	to	08/16/13	08/19/13	to	08/23/13						17944	14784	2	70	2	2.5
07/25/13	to	08/23/13	08/26/13	to	08/30/13						17944	14784	1	30	2.25	2.75
08/01/13	to	08/30/13	09/02/13	to	09/06/13						17944	14784	2	70	2	2.5
08/08/13	to	09/06/13	09/09/13	to	09/13/13	(25)	(45)	1	-25	-25	17919	14739	4	60	1.5	1.75
08/15/13	to	09/13/13	09/16/13	to	09/20/13	763	683	4	-50	-50	18682	15422	4	60	1.25	1.75
08/22/13	to	09/20/13	09/23/13	to	09/27/13						18682	15422	1	20	2.5	1.5
08/29/13	to	09/27/13	09/30/13	to	10/04/13						18682	15422	4	20	2.75	2.5
09/05/13	to	10/04/13	10/07/13	to	10/11/13						18682	15422	4	20	2.75	2.5
09/12/13	to	10/11/13	10/14/13	to	10/18/13						18682	15422	4	20	1.5	2.5
09/19/13	to	10/18/13	10/21/13	to	10/25/13	275	235	2	-138	-138	18957	15657	2	60	1	1.25
09/26/13	to	10/25/13	10/28/13	to	11/01/13	(200)	(220)	1	-200	-200	18757	15437	1	60	1.5	1.25
10/03/13	to	11/01/13	11/04/13	to	11/08/13	2038	1958	4	0	0	20795	17395	4	40	1.5	2.5
10/10/13	to	11/08/13	11/11/13	to	11/15/13	(38)	(58)	1	-38	-38	20757	17337	4	40	1.5	2.5
10/17/13	to	11/15/13	11/18/13	to	11/22/13	713	693	1	0	0	21470	18030	4	40	2	2.5
10/24/13	to	11/22/13	11/25/13	to	11/29/13						21470	18030	2	50	2.5	3.25
10/31/13	to	11/29/13	12/02/13	to	12/06/13	688	668	1	0	0	22158	18698	2	50	2.5	3.25
11/07/13	to	12/06/13	12/09/13	to	12/13/13						22158	18698	3	70	3	3.25
11/14/13	to	12/13/13	12/16/13	to	12/20/13	850	830	1	0	0	23008	19528	3	40	1.75	2
11/21/13	to	12/20/13	12/23/13	to	12/27/13						23008	19528	1	20	2.5	1.5
11/28/13	to	12/27/13	12/30/13	to	01/03/14						23008	19528	1	20	2.5	1.5
12/05/13	to	01/03/14	01/06/14	to	01/10/14	(63)	(103)	2	-163	-163	22945	19425	1	20	2.5	1.5
12/12/13	to	01/10/14	01/13/14	to	01/17/14						22945	19425	3	60	1.5	2.5
12/19/13	to	01/17/14	01/20/14	to	01/24/14						22945	19425	3	60	1.5	2.5
12/26/13	to	01/24/14	01/27/14	to	01/31/14						22945	19425	3	60	1.5	2.5
01/02/14	to	01/31/14	02/03/14	to	02/07/14	563	523	2	0	0	23508	19948	3	60	1.5	2.5
01/09/14	to	02/07/14	02/10/14	to	02/14/14						23508	19948	3	50	3	2.25
01/16/14	to	02/14/14	02/17/14	to	02/21/14						23508	19948	1	50	1.25	1.25
01/23/14	to	02/21/14	02/24/14	to	02/28/14						23508	19948	2	40	1.5	1.5
01/30/14	to	02/28/14	03/03/14	to	03/07/14	550	510	2	-213	-213	24058	20458	2	40	1.5	1.5
02/06/14	to	03/07/14	03/10/14	to	03/14/14	(50)	(70)	1	-50	-50	24008	20388	3	60	2	1.75
02/13/14	to	03/14/14	03/17/14	to	03/21/14	238	198	2	-213	-213	24246	20586	2	50	1.25	2.25
02/20/14	to	03/21/14	03/24/14	to	03/28/14						24246	20586	3	30	1.5	3
02/27/14	to	03/28/14	03/31/14	to	04/04/14	(1088)	(1108)	1	-1088	-1088	23158	19478	3	30	1.5	3
03/06/14	to	04/04/14	04/07/14	to	04/11/14						23158	19478	4	70	3	1.75

In-Sample Date			Out-of-Sample Dates			osnp	NOnp\$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
03/13/14	to	04/11/14	04/14/14	to	04/18/14						23158	19478	4	50	2.25	1.75
03/20/14	to	04/18/14	04/21/14	to	04/25/14						23158	19478	3	50	1.75	1.5
03/27/14	to	04/25/14	04/28/14	to	05/02/14						23158	19478	2	50	1.5	1.75
04/03/14	to	05/02/14	05/05/14	to	05/09/14	450	430	1	0	0	23608	19908	1	60	1.25	2
04/10/14	to	05/09/14	05/12/14	to	05/16/14						23608	19908	1	60	1.25	1.5
04/17/14	to	05/16/14	05/19/14	to	05/23/14						23608	19908	1	60	1.25	1.5
04/24/14	to	05/23/14	05/26/14	to	05/30/14						23608	19908	1	60	1.25	1.5
05/01/14	to	05/30/14	06/02/14	to	06/06/14	838	818	1	0	0	24446	20726	1	60	1.25	1.5
05/08/14	to	06/06/14	06/09/14	to	06/13/14						24446	20726	4	40	2.25	2.5
05/15/14	to	06/13/14	06/16/14	to	06/20/14						24446	20726	4	50	1.75	2.5
05/22/14	to	06/20/14	06/23/14	to	06/27/14						24446	20726	4	50	1.75	2.5
05/29/14	to	06/27/14	06/30/14	to	07/04/14						24446	20726	4	50	1.75	2.5
06/05/14	to	07/04/14	07/07/14	to	07/11/14						24446	20726	4	50	1.75	2.5
06/12/14	to	07/11/14	07/14/14	to	07/18/14	(363)	(383)	1	-363	-363	24083	20343	1	40	1.25	1
06/19/14	to	07/18/14	07/21/14	to	07/25/14	(550)	(610)	3	-250	-550	23533	19733	1	50	0.25	1.25
06/26/14	to	07/25/14	07/28/14	to	08/01/14	(275)	(355)	4	-163	-275	23258	19378	3	70	0.75	1
07/03/14	to	08/01/14	08/04/14	to	08/08/14						23258	19378	3	70	0.75	1.5
07/10/14	to	08/08/14	08/11/14	to	08/15/14	550	490	3	0	0	23808	19868	3	70	0.75	1.25
07/17/14	to	08/15/14	08/18/14	to	08/22/14	25	5	1	0	0	23833	19873	3	70	1.5	1.25
07/24/14	to	08/22/14	08/25/14	to	08/29/14						23833	19873	3	70	1	1.25
07/31/14	to	08/29/14	09/01/14	to	09/05/14	1263	1243	1	0	0	25096	21116	2	60	1.75	0.75
08/07/14	to	09/05/14	09/08/14	to	09/12/14						25096	21116	1	30	1.75	1.25
08/14/14	to	09/12/14	09/15/14	to	09/19/14	250	210	2	-275	-275	25346	21326	1	30	1.25	1.25
08/21/14	to	09/19/14	09/22/14	to	09/26/14	(100)	(160)	3	-200	-200	25246	21166	3	20	1.75	1
08/28/14	to	09/26/14	09/29/14	to	10/03/14	(1313)	(1433)	6	-363	-1475	23933	19733	4	30	2	1.25
09/04/14	to	10/03/14	10/06/14	to	10/10/14	363	343	1	0	0	24296	20076	2	20	2.5	2.25
09/11/14	to	10/10/14	10/13/14	to	10/17/14	1100	1080	1	0	0	25396	21156	1	40	1.5	3.25
09/18/14	to	10/17/14	10/20/14	to	10/24/14						25396	21156	1	30	1.5	3.5
09/25/14	to	10/24/14	10/27/14	to	10/31/14	50	10	2	-125	-125	25446	21166	1	30	1.5	3.5
10/02/14	to	10/31/14	11/03/14	to	11/07/14	(388)	(428)	2	-463	-463	25058	20738	3	60	2.25	2
10/09/14	to	11/07/14	11/10/14	to	11/14/14						25058	20738	3	30	3	2.5
10/16/14	to	11/14/14	11/17/14	to	11/21/14						25058	20738	4	50	2.75	2.25
10/23/14	to	11/21/14	11/24/14	to	11/28/14						25058	20738	4	50	2.75	2.25
10/30/14	to	11/28/14	12/01/14	to	12/05/14	(250)	(310)	3	-225	-250	24808	20428	1	70	1.75	1
11/06/14	to	12/05/14	12/08/14	to	12/12/14						24808	20428	4	20	1.75	2.75
11/13/14	to	12/12/14	12/15/14	to	12/19/14	788	728	3	-175	-250	25596	21156	4	70	2.5	2
11/20/14	to	12/19/14	12/22/14	to	12/26/14						25596	21156	1	20	3.5	2.25
11/27/14	to	12/26/14	12/29/14	to	01/02/15						25596	21156	1	20	3.5	2.25
12/04/14	to	01/02/15	01/05/15	to	01/09/15						25596	21156	1	20	3.5	2.25
12/11/14	to	01/09/15	01/12/15	to	01/16/15	975	835	7	-863	-1213	26571	21991	4	30	2.25	0.25
12/18/14	to	01/16/15	01/19/15	to	01/23/15	2563	2503	3	0	0	29134	24494	2	70	1.75	1.75
12/25/14	to	01/23/15	01/26/15	to	01/30/15	(425)	(445)	1	-425	-425	28709	24049	2	60	2	2
01/01/15	to	01/30/15	02/02/15	to	02/06/15	650	630	1	0	0	29359	24679	2	60	2.5	2
01/08/15	to	02/06/15	02/09/15	to	02/13/15						29359	24679	2	50	2.75	2.25
01/15/15	to	02/13/15	02/16/15	to	02/20/15						29359	24679	2	50	2.75	2.25
01/22/15	to	02/20/15	02/23/15	to	02/27/15	763	723	2	0	0	30122	25402	1	20	1.5	1.5
01/29/15	to	02/27/15	03/02/15	to	03/06/15	600	560	2	0	0	30722	25962	1	20	3.25	1.75
02/05/15	to	03/06/15	03/09/15	to	03/13/15						30722	25962	2	70	2.25	2.75
02/12/15	to	03/13/15	03/16/15	to	03/20/15	1400	1380	1	0	0	32122	27342	1	30	2.25	3
02/19/15	to	03/20/15	03/23/15	to	03/27/15						32122	27342	3	50	3	3.5
02/26/15	to	03/27/15	03/30/15	to	04/03/15	(75)	(95)	1	-75	-75	32047	27247	2	50	2.25	3.25
03/05/15	to	04/03/15	04/06/15	to	04/10/15	(588)	(608)	1	-588	-588	31459	26639	1	40	1.75	3.25
03/12/15	to	04/10/15	04/13/15	to	04/17/15						31459	26639	1	50	3.25	2.25
03/19/15	to	04/17/15	04/20/15	to	04/24/15						31459	26639	3	70	3.5	2.25

In-Sample Date			Out-of-Sample Dates			osnp	NOnp\$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
03/26/15	to	04/24/15	04/27/15	to	05/01/15	263	223	2	-50	-50	31722	26862	3	30	3.5	2.75
04/02/15	to	05/01/15	05/04/15	to	05/08/15	(188)	(328)	7	-825	-1363	31534	26534	4	40	1.75	3.25
04/09/15	to	05/08/15	05/11/15	to	05/15/15						31534	26534	4	30	3.5	3.25
04/16/15	to	05/15/15	05/18/15	to	05/22/15						31534	26534	2	70	1.25	3.25
04/23/15	to	05/22/15	05/25/15	to	05/29/15						31534	26534	1	70	2.75	1.75
04/30/15	to	05/29/15	06/01/15	to	06/05/15	(175)	(235)	3	-350	-350	31359	26299	4	70	3.5	3.25
05/07/15	to	06/05/15	06/08/15	to	06/12/15						31359	26299	3	30	2.75	3.25
05/14/15	to	06/12/15	06/15/15	to	06/19/15	613	573	2	0	0	31972	26872	2	30	1.75	2.25
05/21/15	to	06/19/15	06/22/15	to	06/26/15	(225)	(325)	5	-313	-313	31747	26547	2	50	0.5	2.5
05/28/15	to	06/26/15	06/29/15	to	07/03/15	(763)	(883)	6	-1288	-2138	30984	25664	4	70	1.25	2.75
06/04/15	to	07/03/15	07/06/15	to	07/10/15	(338)	(358)	1	-338	-338	30646	25306	3	60	1.25	3.25
06/11/15	to	07/10/15	07/13/15	to	07/17/15	(200)	(240)	2	-200	-200	30446	25066	4	70	1.5	3.5
06/18/15	to	07/17/15	07/20/15	to	07/24/15						30446	25066	4	20	3	2.75
06/25/15	to	07/24/15	07/27/15	to	07/31/15	238	158	4	-475	-600	30684	25224	4	20	3	2
07/02/15	to	07/31/15	08/03/15	to	08/07/15	(1475)	(1515)	2	-1013	-1475	29209	23709	3	70	3	1.75
07/09/15	to	08/07/15	08/10/15	to	08/14/15						29209	23709	4	40	2	1.75
07/16/15	to	08/14/15	08/17/15	to	08/21/15	413	393	1	0	0	29622	24102	2	20	3	2
07/23/15	to	08/21/15	08/24/15	to	08/28/15	(200)	(220)	1	-200	-200	29422	23882	4	60	2	2.75
07/30/15	to	08/28/15	08/31/15	to	09/04/15	313	253	3	-63	-63	29735	24135	4	60	2	2.75
08/06/15	to	09/04/15	09/07/15	to	09/11/15	225	205	1	0	0	29960	24340	4	20	1.5	2
08/13/15	to	09/11/15	09/14/15	to	09/18/15	625	565	3	-1325	-1325	30585	24905	4	70	1	3
08/20/15	to	09/18/15	09/21/15	to	09/25/15						30585	24905	2	40	2	2.25
08/27/15	to	09/25/15	09/28/15	to	10/02/15	1588	1548	2	0	0	32173	26453	2	40	2	3
09/03/15	to	10/02/15	10/05/15	to	10/09/15						32173	26453	4	30	3.5	3.25
09/10/15	to	10/09/15	10/12/15	to	10/16/15						32173	26453	2	70	2.25	2.25
09/17/15	to	10/16/15	10/19/15	to	10/23/15	1788	1768	1	0	0	33961	28221	2	70	2.25	2.25
09/24/15	to	10/23/15	10/26/15	to	10/30/15	550	530	1	0	0	34511	28751	2	70	3	2.25
10/01/15	to	10/30/15	11/02/15	to	11/06/15	(413)	(453)	2	-375	-413	34098	28298	2	20	2.5	2.75
10/08/15	to	11/06/15	11/09/15	to	11/13/15						34098	28298	2	60	2	1.75
10/15/15	to	11/13/15	11/16/15	to	11/20/15	613	513	5	-488	-488	34711	28811	1	60	1.75	0.25
10/22/15	to	11/20/15	11/23/15	to	11/27/15						34711	28811	4	20	2.5	3
10/29/15	to	11/27/15	11/30/15	to	12/04/15	3000	2960	2	-563	-563	37711	31771	1	60	1.25	1.75
11/05/15	to	12/04/15	12/07/15	to	12/11/15						37711	31771	1	30	2.25	2.75
11/12/15	to	12/11/15	12/14/15	to	12/18/15	(413)	(433)	1	-413	-413	37298	31338	4	60	3.25	3
11/19/15	to	12/18/15	12/21/15	to	12/25/15	200	180	1	0	0	37498	31518	4	40	1	3.25
11/26/15	to	12/25/15	12/28/15	to	01/01/16						37498	31518	3	60	3.25	3.25
12/03/15	to	01/01/16	01/04/16	to	01/08/16						37498	31518	3	60	3.25	3.25
12/10/15	to	01/08/16	01/11/16	to	01/15/16						37498	31518	4	30	1.5	3.5
12/17/15	to	01/15/16	01/18/16	to	01/22/16	1088	1048	2	0	0	38586	32566	4	40	1.75	3
12/24/15	to	01/22/16	01/25/16	to	01/29/16						38586	32566	3	60	2.25	3.25
12/31/15	to	01/29/16	02/01/16	to	02/05/16	956	936	1	0	0	39542	33502	3	60	2.25	3.25
01/07/16	to	02/05/16	02/08/16	to	02/12/16						39542	33502	4	40	3.5	3.5
01/14/16	to	02/12/16	02/15/16	to	02/19/16						39542	33502	2	60	2.25	3.5
01/21/16	to	02/19/16	02/22/16	to	02/26/16						39542	33502	2	60	2.25	3.5
01/28/16	to	02/26/16	02/29/16	to	03/04/16	1463	1383	4	0	0	41005	34885	4	20	2	2.25
02/04/16	to	03/04/16	03/07/16	to	03/11/16	1600	1440	8	-931	-950	42605	36325	4	20	2	3.25
02/11/16	to	03/11/16	03/14/16	to	03/18/16	1088	1068	1	0	0	43693	37393	3	20	2.25	2.75
02/18/16	to	03/18/16	03/21/16	to	03/25/16						43693	37393	2	30	3	3.5
02/25/16	to	03/25/16	03/28/16	to	04/01/16	300	260	2	-294	-294	43993	37653	2	70	1.25	2.75
03/03/16	to	04/01/16	04/04/16	to	04/08/16						43993	37653	3	70	2.5	3.25
03/10/16	to	04/08/16	04/11/16	to	04/15/16						43993	37653	1	20	2.75	1.75
03/17/16	to	04/15/16	04/18/16	to	04/22/16	(1156)	(1176)	1	-1156	-1156	42837	36477	4	70	1.5	2
03/24/16	to	04/22/16	04/25/16	to	04/29/16	(388)	(408)	1	-388	-388	42449	36069	4	30	3.25	2
03/31/16	to	04/29/16	05/02/16	to	05/06/16	(694)	(754)	3	-281	-694	41755	35315	2	40	1.75	1.5

In-Sample Date			Out-of-Sample Dates			osnp	NOnp\$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
04/07/16	to	05/06/16	05/09/16	to	05/13/16	(50)	(70)	1	-50	-50	41705	35245	1	70	1.25	1
04/14/16	to	05/13/16	05/16/16	to	05/20/16	356	336	1	0	0	42061	35581	1	40	1.5	0.75
04/21/16	to	05/20/16	05/23/16	to	05/27/16						42061	35581	3	50	2.5	2
04/28/16	to	05/27/16	05/30/16	to	06/03/16	1425	1405	1	0	0	43486	36986	2	50	1.5	1.75
05/05/16	to	06/03/16	06/06/16	to	06/10/16						43486	36986	2	60	1.25	2.5
05/12/16	to	06/10/16	06/13/16	to	06/17/16	(119)	(159)	2	-319	-319	43367	36827	2	60	1.25	2.5
05/19/16	to	06/17/16	06/20/16	to	06/24/16	(588)	(628)	2	-575	-588	42779	36199	1	30	2.5	1.25
05/26/16	to	06/24/16	06/27/16	to	07/01/16	106	86	1	0	0	42885	36285	1	70	1.5	1.25
06/02/16	to	07/01/16	07/04/16	to	07/08/16						42885	36285	1	70	1.5	1.75
06/09/16	to	07/08/16	07/11/16	to	07/15/16						42885		36285			
06/16/16	to	07/15/16	07/18/16	to	07/22/16						42885	36285	3	60	1.5	3
06/23/16	to	07/22/16	07/25/16	to	07/29/16	494	474	1	0	0	43379	36759	4	30	1.5	2.25
06/30/16	to	07/29/16	08/01/16	to	08/05/16	(31)	(71)	2	-25	-31	43348	36688	4	70	1.5	3
07/07/16	to	08/05/16	08/08/16	to	08/12/16	(444)	(464)	1	-444	-444	42904	36224	2	40	1.25	3.25
07/14/16	to	08/12/16	08/15/16	to	08/19/16	6	(54)	3	-138	-213	42910	36170	3	50	2	1.25
07/21/16	to	08/19/16	08/22/16	to	08/26/16	31	(29)	3	-1019	-1350	42941	36141	3	50	2.25	1.25
07/28/16	to	08/26/16	08/29/16	to	09/02/16	481	461	1	0	0	43422	36602	3	70	2.25	1.75
08/04/16	to	09/02/16	09/05/16	to	09/09/16						43422	36602	4	30	3	3
08/11/16	to	09/09/16	09/12/16	to	09/16/16						43422	36602	2	40	2.75	1.75
08/18/16	to	09/16/16	09/19/16	to	09/23/16						43422	36602	4	50	2.75	2.25
08/25/16	to	09/23/16	09/26/16	to	09/30/16						43422	36602	4	50	2.75	2.25
09/01/16	to	09/30/16	10/03/16	to	10/07/16						43422	36602	1	60	2.25	1
09/08/16	to	10/07/16	10/10/16	to	10/14/16						43422	36602	3	20	2	2.25
09/15/16	to	10/14/16	10/17/16	to	10/21/16	744	704	2	-200	-200	44166	37306	3	20	2	2.25
09/22/16	to	10/21/16	10/24/16	to	10/28/16						44166	37306	4	50	3	1.75
09/29/16	to	10/28/16	10/31/16	to	11/04/16						44166	37306	4	50	3	1.75
10/06/16	to	11/04/16	11/07/16	to	11/11/16	606	586	1	0	0	44772	37892	4	50	3	1.75
10/13/16	to	11/11/16	11/14/16	to	11/18/16	894	854	2	-250	-250	45666	38746	4	30	2	1.25
10/20/16	to	11/18/16	11/21/16	to	11/25/16						45666	38746	3	30	2	1.5
10/27/16	to	11/25/16	11/28/16	to	12/02/16	(569)	(629)	3	-313	-569	45097	38117	4	30	1.75	1.25
11/03/16	to	12/02/16	12/05/16	to	12/09/16	1713	1693	1	0	0	46810	39810	1	20	1.5	1.5
11/10/16	to	12/09/16	12/12/16	to	12/16/16	838	818	1	0	0	47648	40628	2	40	2	1.25
11/17/16	to	12/16/16	12/19/16	to	12/23/16						47648	40628	4	40	3.25	2.5
11/24/16	to	12/23/16	12/26/16	to	12/30/16						47648	40628	4	40	3.25	2.5
12/01/16	to	12/30/16	01/02/17	to	01/06/17	(56)	(96)	2	-344	-344	47592	40532	4	40	3.25	1.5
12/08/16	to	01/06/17	01/09/17	to	01/13/17	756	716	2	-238	-238	48348	41248	4	40	2.5	2.25
12/15/16	to	01/13/17	01/16/17	to	01/20/17	(569)	(589)	1	-569	-569	47779	40659	2	70	1.25	2
12/22/16	to	01/20/17	01/23/17	to	01/27/17						47779	40659	4	40	2.5	2.75
12/29/16	to	01/27/17	01/30/17	to	02/03/17						47779	40659	4	40	2.5	2.75
01/05/17	to	02/03/17	02/06/17	to	02/10/17						47779	40659	3	70	0.75	3.25
01/12/17	to	02/10/17	02/13/17	to	02/17/17	1094	1034	3	0	0	48873	41693	3	70	0.75	1.75
01/19/17	to	02/17/17	02/20/17	to	02/24/17	494	454	2	-38	-38	49367	42147	3	70	0.75	1.5
01/26/17	to	02/24/17	02/27/17	to	03/03/17	225	205	1	0	0	49592	42352	2	70	1	1.25
02/02/17	to	03/03/17	03/06/17	to	03/10/17	300	260	2	0	0	49892	42612	4	60	1.25	2
02/09/17	to	03/10/17	03/13/17	to	03/17/17	319	259	3	-75	-75	50211	42871	4	20	1.25	2
02/16/17	to	03/17/17	03/20/17	to	03/24/17						50211	42871	2	20	1	3
02/23/17	to	03/24/17	03/27/17	to	03/31/17						50211	42871	3	60	1.25	2
03/02/17	to	03/31/17	04/03/17	to	04/07/17						50211	42871	4	70	1.25	2
03/09/17	to	04/07/17	04/10/17	to	04/14/17	394	374	1	0	0	50605	43245	4	70	1.25	2
03/16/17	to	04/14/17	04/17/17	to	04/21/17						50605	43245	1	60	1	1.5
03/23/17	to	04/21/17	04/24/17	to	04/28/17	(469)	(549)	4	-556	-556	50136	42696	2	50	1	1.5
03/30/17	to	04/28/17	05/01/17	to	05/05/17	125	45	4	-225	-263	50261	42741	2	20	0.75	0.75
04/06/17	to	05/05/17	05/08/17	to	05/12/17	256	236	1	0	0	50517	42977	1	20	1	1.5
04/13/17	to	05/12/17	05/15/17	to	05/19/17	(275)	(295)	1	-275	-275	50242	42682	1	70	1	0.75

In-Sample Date			Out-of-Sample Dates			osnp	NOnp\$20	ont	ollt	odd	EQ	NetEq	pw	N	vup	vdn
04/20/17	to	05/19/17	05/22/17	to	05/26/17						50242	42682	1	70	1	1.5
04/27/17	to	05/26/17	05/29/17	to	06/02/17	206	186	1	0	0	50448	42868	1	70	1	1.5
05/04/17	to	06/02/17	06/05/17	to	06/09/17	131	111	1	0	0	50579	42979	3	70	1	2
05/11/17	to	06/09/17	06/12/17	to	06/16/17	(325)	(345)	1	-325	-325	50254	42634	3	40	1.25	2
05/18/17	to	06/16/17	06/19/17	to	06/23/17						50254	42634	4	30	1.75	1.5
05/25/17	to	06/23/17	06/26/17	to	06/30/17	281	201	4	-300	-369	50535	42835	4	40	2	1.5
06/01/17	to	06/30/17	07/03/17	to	07/07/17						50535	42835	4	30	3	3
06/08/17	to	07/07/17	07/10/17	to	07/14/17						50535	42835	4	30	1.75	1.5
06/15/17	to	07/14/17	07/17/17	to	07/21/17						50535	42835	4	30	1.5	2
06/22/17	to	07/21/17	07/24/17	to	07/28/17	494	474	1	0	0	51029	43309	3	60	1.25	3.25
06/29/17	to	07/28/17	07/31/17	to	08/04/17	6	(34)	2	-581	-581	51035	43275	1	20	1.25	2.25
07/06/17	to	08/04/17	08/07/17	to	08/11/17	(869)	(929)	3	-781	-869	50166	42346	3	40	1.5	2
07/13/17	to	08/11/17	08/14/17	to	08/18/17	1513	1413	5	-88	-88	51679	43759	1	70	0.25	2.5
07/20/17	to	08/18/17	08/21/17	to	08/25/17	1338	1238	5	-100	-100	53017	44997	1	70	0.25	2.75
07/27/17	to	08/25/17	08/28/17	to	09/01/17	125	85	2	-613	-613	53142	45082	2	40	1.75	2.25
08/03/17	to	09/01/17	09/04/17	to	09/08/17						53142	45082	3	20	3.5	2.5
08/10/17	to	09/08/17	09/11/17	to	09/15/17	(619)	(639)	1	-619	-619	52523	44443	3	40	3	2.25
08/17/17	to	09/15/17	09/18/17	to	09/22/17	869	849	1	0	0	53392	45292	3	40	3	2.75
08/24/17	to	09/22/17	09/25/17	to	09/29/17						53392	45292	3	50	3	2.75

Appendix I: n^{th} Order Polynomial Next Bar's Forecast Math

What is the n^{th} Order Polynomial ?

The n^{th} Order Polynomial, also called the n^{th} Order Fixed Memory Polynomial, is simply the least square fit of a polynomial of the form $\mathbf{b}_0 + \mathbf{b}_1 * \mathbf{t} + \mathbf{b}_2 * \mathbf{t}^2 + \mathbf{b}_3 * \mathbf{t}^3 + \dots + \mathbf{b}_n * \mathbf{t}^n$ to a *fixed* number of past data points. Where \mathbf{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 $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$ for the polynomial is determined. This type of error minimization is mathematically solvable and is widely used in science and mathematics.

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

$$\begin{bmatrix} \mathbf{T} & \sum \mathbf{t} & \sum \mathbf{t}^2 & \sum \mathbf{t}^3 & \sum \mathbf{t}^4 \\ \sum \mathbf{t} & \sum \mathbf{t}^2 & \sum \mathbf{t}^3 & \sum \mathbf{t}^4 & \sum \mathbf{t}^5 \\ \sum \mathbf{t}^2 & \sum \mathbf{t}^3 & \sum \mathbf{t}^4 & \sum \mathbf{t}^5 & \sum \mathbf{t}^6 \\ \sum \mathbf{t}^3 & \sum \mathbf{t}^4 & \sum \mathbf{t}^5 & \sum \mathbf{t}^6 & \sum \mathbf{t}^7 \\ \sum \mathbf{t}^4 & \sum \mathbf{t}^5 & \sum \mathbf{t}^6 & \sum \mathbf{t}^7 & \sum \mathbf{t}^8 \end{bmatrix} \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{b}_0 \\ \mathbf{c}_0 \\ \mathbf{d}_0 \\ \mathbf{e}_0 \end{bmatrix} = \begin{bmatrix} \sum \mathbf{p}(\mathbf{t}) \\ \sum (\mathbf{p}(\mathbf{t}) * \mathbf{t}) \\ \sum (\mathbf{p}(\mathbf{t}) * \mathbf{t}^2) \\ \sum (\mathbf{p}(\mathbf{t}) * \mathbf{t}^3) \\ \sum (\mathbf{p}(\mathbf{t}) * \mathbf{t}^4) \end{bmatrix}$$

where

$\mathbf{p}(\mathbf{T})$ is the current bar's price, $\mathbf{p}(\mathbf{T}-1)$ is the previous bar's price and $\mathbf{p}(\mathbf{1})$ is the price \mathbf{T} bars ago.

\mathbf{T} is the number of Bars in the Least Squares estimation

$\sum \mathbf{p}(\mathbf{t})$ is the summation of prices from $\mathbf{t}=1$ to \mathbf{T} bars

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

$\sum \mathbf{t}$ is the summation of the integer \mathbf{t} from $\mathbf{t}=1$ to \mathbf{T} bars

$\sum \mathbf{t}^2$ is the summation of the integer \mathbf{t} squared from $\mathbf{t}=1$ to \mathbf{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:

$$\mathbf{P}_f = \mathbf{a}_0 + \mathbf{b}_0 * (\mathbf{T}+1) + \mathbf{c}_0 * (\mathbf{T}+1)^2 + \mathbf{d}_0 * (\mathbf{T}+1)^3 + \mathbf{e}_0 * (\mathbf{T}+1)^4$$

Where \mathbf{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:

$$\mathbf{Velocity}(\mathbf{T}+1) = \mathbf{dP}_f / \mathbf{dt} = \mathbf{b}_0 + 2\mathbf{c}_0 * (\mathbf{T}+1) + 3\mathbf{d}_0 * (\mathbf{T}+1)^2 + 4\mathbf{e}_0 * (\mathbf{T}+1)^3$$

$$\mathbf{Velocity}(\mathbf{t}+1) = \mathbf{d}^2 \mathbf{P}_f / \mathbf{d}^2 \mathbf{t} = 2 \mathbf{c}_0 + 6\mathbf{d}_0 * (\mathbf{T}+1) + 12\mathbf{e}_0 * (\mathbf{T}+1)^2$$

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

Orthogonal 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, $\beta_0, \beta_1, \beta_2, \beta_3, \dots, \beta_n$ for a n^{th} 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

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

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

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

The n^{th} Order Fixed Memory Forecast Next Bar's Velocity Strategy Defined

The least squares forecast is constructed by solving for the least squares coefficients $\beta_1, \beta_2, \dots, \beta_n$ 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 $b_0 + b_1*t + b_2*t^2 + b_3*t^3 + \dots + b_n*t^n$ 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 $b_0, b_1, b_2, \dots, b_n$ for the polynomial is determined. This type of error minimization is mathematically solvable and is widely used in science and mathematics. Once the b_n coefficients are found using a lookback period of T bars to calculate the 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.

$$\text{Velocity}(T+1) = dP_t/dt = b_0 + 2c_0*(T+1) + 3d_0*(T+1)^2 + 4e_0*(T+1)^3 + \dots + n*b_n*(T+1)^{n-1}$$

Please see the *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

The Normalization Multiplier

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 than 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=1.3322 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

The *Normalization Multiplier*

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 *independent* 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