

The n^{th} Order Polynomial Acceleration Strategy Applied to 24hr Euro Dollar Futures 1min Bars

Working Paper July 2012
Copyright © 2012 Dennis Meyers

In previous working papers we showed how the application of a price curve generated by second, third and fourth degree polynomial velocity could be used to develop a strategy to buy and sell intraday futures. The reasoning behind this type of system was to only trade when the price trend velocity was above or below a certain threshold. Many times prices are meandering around without a notable trend. At these times we do not wish to trade because of the whipsaws losses that occur from this type of price action. When a price trend finally starts, the velocity of that price trend moves above some minimum value. Thus the velocity system would only issue a trade when certain velocity thresholds were crossed. Acceleration is the change in velocity per unit time. In order for velocity to reach a certain value the price changes must start accelerating. Thus acceleration should be an even earlier indicator than velocity that prices are starting to trend. Here we will use walk forward optimization to determine the “best” polynomial order and use the Fixed Memory Polynomial Acceleration System to trade the 24hr Euro Dollar futures contract on an intraday basis using one minute bar price data. To test this system we will use one minute bar prices of the Eurodollar futures contract traded on the CME and known by the symbol EC for the two years from June 25, 2010 to June 22, 2012.

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 least squares coefficients a_0, b_0, c_0, d_0, e_0 each day using the last N bars of closing prices as shown in the “Math” appendix at the end of this working paper. Then **Acceleration(T+1)** is constructed from the equation shown in the “Math” appendix and plotted under the price chart.

The acceleration of a 2nd, 3rd and 4th order polynomial curve should change faster than its corresponding polynomial curve velocity. Whether higher order polynomial accelerations are an advantage or not, we will let the computer decide when we use a walk forward optimization technique described below.

At each bar we calculate the n^{th} order (2nd through 4th) fixed memory polynomial acceleration from the formulas in the “Math” appendix. As we will show below, optimization will determine the order for n^{th} order polynomial acceleration that will be used. When the acceleration is greater than the threshold amount **aup** we will go long. When the acceleration is less than the threshold amount **-adn** we will go short.

Buy Rule:

IF Acceleration is greater than the threshold amount **aup** then buy at the market.

Sell Rule:

IF Acceleration is less than the threshold amount **-adn** then sell at the market.

Discussion of Eurodollar Prices

The Eurodollar(EC) is traded on Globex and on the trading floor at the CME. On Globex the EC is traded on a 23hour basis. The CME hours for floor trading (RTH) are 7:20 to 14:00 CST. Over 50% of the volume in the EC is done on Globex is during the CME RTH hours. However for this paper we will trade all hours that Globex is open

Testing The Polynomial Acceleration System Using Walk Forward Optimization

There will be four strategy parameters to determine:

1. **degree**, degree=2 for 2nd order acceleration, degree=3 for 3rd order acceleration, degree=4 for 4th order acceleration,
2. N , is the number of lookback bars of prices to calculate the **acceleration**.
3. **aup**, the threshold amount that acceleration has to be greater than to issue a buy signal
4. **adn**, the threshold amount that acceleration has to be less than to issue a sell signal

To test this system we will use one minute bar prices of the Eurodollar(EC) futures contract traded on the CME and known by the symbol EC from June 25, 2010 to June 22, 2012

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 n^{th} Order Polynomial Acceleration Strategy on the EC 1min data from 6/25/2010 to 6/22/2012. The PWFO will breakup and create 105 30 calendar day in-sample/test sections along with their corresponding 7 calendar day out-of-sample sections from the two years of EC TS optimization (see Walk forward Testing below). The test and out-of-sample section dates are shown in **Table 1** on page 7 below. We will then use another software product called the Walk Forward Metric Explorer (WFME) on each of the 152 test and out-of-sample(oos) sections generated by the PWFO to find the best in-sample/test section performance **filter** that determines the system input parameters (**degree, N, aup, adn**) that will be used on the out-of-sample data. Detailed information about the PWFO and the WFPE can be found at www.meyersanalytics.com

For the test data we will run the TradeStation optimization engine on the two years of EC 1 min bars with the following ranges for the nth order polynomial acceleration strategy inputs variables.

1. degree from 2 to 4
2. N from 20 to 70 in steps of 10.
3. aup from 0.2 to 2.4 steps of 0.2
4. adn from 0.2 to 2.4 in steps of 0.2

This will produce 2592 different cases or combinations of the input parameters for each of the 105 PWFO output files.

Walk Forward Out-of-Sample Testing

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

Why use the walk forward technique? Why not just perform an optimization on the whole price series and choose the input parameters that give the best total net profits or profit factor? Surely the price noise cancels itself out with such a large number of test prices and trades. Unfortunately, nothing could be farther from the truth! Optimization is a misnomer and should really be called combinatorial search. As stated above, whenever we run a combinatorial search over many different combinations of input parameters or even with genetic(Grail) searches on noisy data on a fixed number of prices, **no matter how many**, the best performance parameters found are guaranteed to be due to "curve fitting" the noise and signal. What do we mean by "curve fitting"? Price series that we trade consists of random spurious price movements, which we call noise, and repeatable price patterns (*if they exist*). When we run, say, 8000 different input parameter combinations, the best performance parameters will be from those system input

variables that are able to produce profits from the price pattern *and* the random spurious movements. While the price patterns will repeat, the same spurious price movements will not. If the spurious movements that were captured by a certain set of input parameters were a large part of the total net profits, as they are in real price data, then choosing these input parameters will produce losses when traded on future data. These losses occur because the spurious movements will not be repeated in the same way. This is why strategy optimization or combinatorial searches with no out-of-sample testing cause losses when traded in real time from something that looked great in in-sample/test section. Unfortunately it is human nature to extrapolate past performance to project future trading results and so results from curve fitting give the illusion, a modern “siren call” so to speak, of future trading profits.

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

Finding The Strategy Input Parameters in The Walk Forward In-sample/test sections

The PWFO generates a number of performance metrics in the in-sample/test section. The question we are attempting to answer statistically, is which performance metric or combination of performance metrics (which we will call a *filter*) in in-sample/test section will produce strategy inputs that produce statistically valid profits in the out-of-sample section. In other words we wish to find a performance metric *filter* that we can apply to the in-sample/test section that can give us strategy inputs that will produce, on average, good trading results in the future. The PWFO produces a total of 32 different performance metrics in in-sample/test section. If we have 2592 different strategy input combinations or cases then the in-sample/test section consists of 32 columns of performance metrics for each of the 2592 input cases or rows.

Appendix A shows an example of a truncated PWFO file for Out-Of-Sample end date of 6/22/12:

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

Here is an *example* of a better more complicated *filter* that was used in this paper. We require that the number of trades (NT) in the in-sample/test section be greater than 10 trades a month. We require this so that we can eliminate strategy inputs that produce very few trades. One calendar month is approximately 21 trading days so we are requiring at least a trade every other day on average. Not many traders can stay with a strategy that has a large number of losers in a row (LR). For this filter we will choose $LR \leq 3$. This choice of LR is completely arbitrary and is what I feel comfortable with. After using a NT and LR filter, as described, there can still be 100's of rows left in the PWFO file in-sample/test section. The PWFO generates the metric *p/l*. This metric calculates the ratio of the **Median Winning Trade Profit divided by the absolute value of the Median of the Losing Trades**. We use the median for this metric, rather than the average trade profit, because we do not want this metric distorted by any outlier trades, that is trades that generate very large profits of losses one or two times. Thus we would want the *p/l* to be as large as possible. Let us choose the 20 rows that contain the **largest(top) p/l** values from the rows that are left from the NT-LR screen. In other words we sort *p/l* from high to low, eliminate the rows that have $NT < 10$ and $LR > 3$ then choose the top 20 Rows of whatever is left. This particular filter will now leave 20 cases or rows in the PWFO file that satisfy the above filter conditions. We call this filter **t20p/l|lr3|nt10** where **t20p/l** means the Top 20 *p/l* rows left *after* the NT-LR filter.. Suppose for this filter, within the 20 PWFO rows that are left, we want the row that has the maximum **median (mp-rd)** metric in the in-sample/test section. The **pwfo metric mp-rd is the final trade profit minus the maximum rundown of the trade. That is for each trade we measure where the trade closed**

compared to its worst loss from the start of the trade. Each set of strategy inputs in the in-sample/test section has number of winning trades and losing trades. Each winning trade and losing trade has a mp-rd value. Each row in the PWFO file represents a set of strategy inputs that has an associated median mp-rd value.. This would produce a filter named **t20p/l|lr3|nt10-mp-rd**. For each **in-sample/test section** this **filter** leaves only one row in the PWFO in-sample/test section with its associated strategy inputs and out-of-sample net profit. This particular **t20p/l|lr3|nt10-mp-rd** is then applied to each of the 105 PWFO in-sample/test sections which gives 105 sets of strategy inputs that are used to produce the corresponding 105 out-of-sample performance results. The average out-of-sample performance is calculated from these 105 out-of-sample performance results. In addition many other important out-of-sample performance statistics for this filter are calculated and summarized. **Figure 3** shows such a computer run along with a small sample of other filter combinations that are constructed in a similar manner. Row 3 of the sample output in Figure 3 shows the results of the filter discussed above.

Bootstrap Probability of Filter Results. Using modern "Bootstrap" techniques, we can calculate the probability of obtaining 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 have 100 PWFO files of test/oos data and we calculate the total out-of-sample net profits(tOnpNet) for 5000 different **TopN-Metric- LR-NT** filters. A mirror filter is created for each of the 5000 filters. However, instead of picking an out-of-sample net profit(OSNP) from a filter row, the mirror filter picks a *random* row's OSNP in each of the 100 PWFO files. Each of the 5000 mirror filters will choose a random row's OSNP of their own in each of the 100 PWFO files.. At the end, each of the 5000 mirror filters will have 100 *random* OSNP's picked from the rows of the 100 PWFO files. The sum of the 100 random OSNP picks for each mirror filter will generate a random total out-of-sample net profit (tOnpNet). The average and standard deviation of the 5000 mirror filter's different random tOnpNets will allow us to calculate the chance probability of each **TopN-Metric- LR-NT** filter's tOnpNet. Thus given the mirror filter's bootstrap random tOnpNet average and standard deviation, we can calculate the probability of obtaining the TopN-Metric-PF-LR-NT filter's tOnpNet by pure chance alone. Since for this run there are 15376 different filters, we can calculate the expected number of cases that we could obtain by pure chance that would match or exceed the tOnpNet of the filter we have chosen or $(15376) \times (\text{tOnpNet Probability})$. For our filter, in row 3 in Figure 3, the expected number of cases that we could obtain by pure chance that would match or exceed the \$72206 is $15376 \times 1.22 \cdot 10^{-6} = 0.0187$. This is much less than one so it is improbable that our result was due to pure chance.

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

Filter: #Trds>=10and LR<=3 and Top 20 p/l then maximum mp-rd

Where:

NT = The number of trades for a given set of strategy inputs in the in-sample/test section.

LR = Maximum losing trades in a row for a given set of strategy inputs in the in-sample/test section

p/l = The ratio of the **Median Winning Trade Profit divided by the absolute value of the Median of the Losing Trades** in the in-sample section

mp-rd = The median of **the final trade profit minus the maximum rundown of the trade** in the in-sample/test section

The first part of the filter chooses those rows or cases out of the 2592 rows in each PWFO file test(in-sample) section that satisfy the criteria **LR<=3 and NT>=10**. After using a NT and LR filter, there can still be 100's of rows left in the PWFO file. The PWFO generates the metric **p/l**, which is the ratio of the median winning trade Profit divided by the absolute value of the median of the losing trades for any given set of strategy inputs in the in-sample/test section. Let us choose out of the rows that are left from the **NT-LR** screen, the **20** rows that contain the **top 20 p/l**. This particular filter will now leave 20 cases or rows in the PWFO file that satisfy these filter conditions. We call this filter **t20p/l|lr3|nt10-mp-rd** where **t20p/l** means the **Top 20 p/l** rows left *after* the NT-LR filter. Within the **20** PWFO rows that are left, we want the row that has the maximum PWFO metric, **mp-rd**, in the in-sample/test section. This **Filter** or selection procedure will leave only one choice for the system input values of *degree, N, aup*,

adn. We then use these input values found in the in-sample/test section by the **Filter** on the **next week** of one minute EC **out-of-sample** price bars **following** the in-sample/test section.

Results

Table 1 below presents a table of the 105 test and out-of-sample windows, the **Filter** selected, strategy inputs and the weekly out-of-sample profit/loss results using the filter described above.

Figure 1 presents a graph of the equity curve generated by using the filter on the 105 weeks of 6/25/10 – 6/25/12. The equity curve is plotted from Equity column in Table 1. Plotted on the equity curve is the least squares straight line.

Figure 2 presents the out-of-sample 1 minute bar chart of EC for 6/4/12 to 6/5/12 with the Nth Order Polynomial Acceleration System Indicator and all the buy and sell signals for those dates.

Figure 3 Partial output of the Walk Forward Metric Performance Explorer

Discussion of System Performance

In Figure 3 Row 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 17 weeks for this filter. This means that 17 weeks was the longest time that the equity for this strategy failed to make a new equity high.

To see the effect of walk forward analysis, take a look at **Table 1**. Notice how the input parameters *degree*, *N*, *aup*, *adn* take sudden jumps from high to low and back. This is the walk forward process quickly adapting to changing volatility conditions in the test sample. In addition, notice how often *degree* changes from 2 to 4 and *aup* changes from 0.4 to 2.4. When the data gets very noisy with a lot of spurious price movements, the degree, has to be lower. During other times when the noise level is not as much degree can be higher to get onboard a trend faster.

In Figure 1, which presents a graph of the equity curve using the filter on the 105 weeks of out-of-sample data, notice how the equity curve follows the trend line with an R^2 of 0.94.

Using this filter, the strategy was able to generate \$72,206 net equity trading one EC contract for 105 weeks. The strategy did not trade in 25 of the 105 weeks. Either the NT or LR or $tnp > 0$ criteria was not met by any set of inputs in the in-sample section. The period of time from 6/25/10 to 6/22/12 was a very volatile market. Yet the Nth Order Polynomial Acceleration strategy was able to adapt quite well. From Table 1, the largest losing week was -\$4813 on the week ending 10/14/11 a very wild financial time and market week. The largest drawdown was -\$6501 from the week ending on 6/17/11 to 10/14/11 during that same wild financial time. However during this 17 week period the strategy did not trade for 13 weeks during this period from 7/15/11 to 10/7/11 and completely recovered and made a new equity high within two weeks on 10/21/12. The longest time between new equity highs was 17 weeks.

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 76 trades/week with an average of 16.6 trades/week on the weeks the strategy traded. The 76 trades /week occurred for the week ending 6/17/11. *degree* was 4, *N* was 60 and *aup* was an unusually low *aup*=0.4. This was a big up week that followed a big down week and caused an unusual number of whipsaw trades on the buy side although the week ended with a profit of \$1515.

Given 23 hour trading of the EC, the strategy did not miss any profitable trends opportunities when Asia and then Europe opened trading in the early morning.

Disclaimer

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

References

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

Table 1 Walk Forward Out-Of-Sample Performance Summary for EC1m Nth Order Polynomial Acceleration System

EC-1 min bars 6/25/2010 - 6/22/2012. The input values *degree(pw)*, *N*, *aup*, *adn* are the values found from applying the filter to the test Sample optimization runs.

Test Section Filter: #Trds>=10 and LR<=3 and Top 20 p/l then maximum mp-rp

Where:

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

LR = Maximum losing trades in a row for a given set of strategy inputs in the test section

p/l = The ratio of the Median Winning Trade Profit divided by the absolute value of the Median of the Losing Trades in the in-sample section.

mp-rp = The median of the final trade profit minus the maximum rundown of the trade in the in-sample/test section

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

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

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

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

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

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

pw = polynomial degree. degree=2 for parabolic curve acceleration, degree=3 for 3rd order acceleration, etc.

N = is the number of bars lookback period to calculate the polynomial **acceleration**.

aup =, the threshold amount that acceleration has to be greater than to issue a buy signal

adn =, the threshold amount that acceleration has to be less than to issue a sell signal

Week#	In-Sample Dates	Out-Of-Sample Dates	Osnp	OnT	ollt	odd	avosnp	Equity	Pw	N	aup	adn
1	05/19/10 To 06/18/10	06/21/10 To 06/25/10	0	0				0				
2	05/26/10 To 06/25/10	06/28/10 To 07/02/10	2900	22	(888)	(1250)	132	2900	2	70	1.2	2.2
3	06/02/10 To 07/02/10	07/05/10 To 07/09/10	1088	1	0	0	1088	3988	2	70	1	2.4
4	06/09/10 To 07/09/10	07/12/10 To 07/16/10	4100	8	(275)	(275)	513	8088	2	70	1	2.4
5	06/16/10 To 07/16/10	07/19/10 To 07/23/10	(925)	22	(1500)	(3400)	(42)	7163	2	70	1.2	2.4
6	06/23/10 To 07/23/10	07/26/10 To 07/30/10	1538	4	(563)	(875)	385	8701	2	70	2	2.4
7	06/30/10 To 07/30/10	08/02/10 To 08/06/10	3000	6	(588)	(588)	500	11701	2	60	2	2.4
8	07/07/10 To 08/06/10	08/09/10 To 08/13/10	3188	11	(863)	(863)	290	14889	2	70	2	2.4
9	07/14/10 To 08/13/10	08/16/10 To 08/20/10	1513	10	(1375)	(1388)	151	16402	2	70	2	2.4
10	07/21/10 To 08/20/10	08/23/10 To 08/27/10	3063	11	(900)	(1025)	279	19465	2	60	1.8	2.4
11	07/28/10 To 08/27/10	08/30/10 To 09/03/10	(2250)	4	(900)	(1250)	(563)	17215	3	70	2.4	2.2
12	08/04/10 To 09/03/10	09/06/10 To 09/10/10	413	5	(688)	(1025)	83	17628	2	50	2.4	2.4
13	08/11/10 To 09/10/10	09/13/10 To 09/17/10	(2500)	13	(2263)	(3325)	(192)	15128	2	50	2.4	2.4
14	08/18/10 To 09/17/10	09/20/10 To 09/24/10	(1038)	10	(1550)	(2413)	(104)	14090	2	40	2.4	2.2
15	08/25/10 To 09/24/10	09/27/10 To 10/01/10	2925	26	(600)	(1388)	113	17015	3	60	1.4	2.4
16	09/01/10 To 10/01/10	10/04/10 To 10/08/10	1825	32	(688)	(1538)	57	18840	2	40	0.8	2.2
17	09/08/10 To 10/08/10	10/11/10 To 10/15/10	875	28	(2250)	(2488)	31	19715	4	70	0.2	2.4
18	09/15/10 To 10/15/10	10/18/10 To 10/22/10	(638)	50	(1300)	(2425)	(13)	19077	3	50	0.2	2.4
19	09/22/10 To 10/22/10	10/25/10 To 10/29/10	0	0				19077				
20	09/29/10 To 10/29/10	11/01/10 To 11/05/10	0	0				19077				
21	10/06/10 To 11/05/10	11/08/10 To 11/12/10	0	0				19077				
22	10/13/10 To 11/12/10	11/15/10 To 11/19/10	800	6	(1750)	(2663)	133	19877	2	50	2.4	1.2
23	10/20/10 To 11/19/10	11/22/10 To 11/26/10	875	28	(913)	(1600)	31	20752	2	50	2.4	1.2
24	10/27/10 To 11/26/10	11/29/10 To 12/03/10	0	0				20752				

Week#	In-Sample Dates		Out-Of-Sample Dates		Osnp	OnT	ollt	odd	avosnp	Equity	Pw	N	aup	adn
25	11/03/10	To 12/03/10	12/06/10	To 12/10/10	4888	9	(363)	(513)	543	25640	2	70	2.4	2.2
26	11/10/10	To 12/10/10	12/13/10	To 12/17/10	6713	10	(763)	(1013)	671	32353	3	70	2.4	2.4
27	11/17/10	To 12/17/10	12/20/10	To 12/24/10	300	8	(475)	(513)	38	32653	3	70	2.4	2.4
28	11/24/10	To 12/24/10	12/27/10	To 12/31/10	(750)	5	(1488)	(1800)	(150)	31903	2	60	2.2	2.4
29	12/01/10	To 12/31/10	01/03/11	To 01/07/11	1138	25	(1838)	(2538)	46	33041	3	70	2.4	2.4
30	12/08/10	To 01/07/11	01/10/11	To 01/14/11	3375	26	(1050)	(1525)	130	36416	2	50	1.8	2.4
31	12/15/10	To 01/14/11	01/17/11	To 01/21/11	1550	27	(700)	(2800)	57	37966	2	50	1.6	2.4
32	12/22/10	To 01/21/11	01/24/11	To 01/28/11	788	12	(525)	(838)	66	38754	2	70	1.6	2.4
33	12/29/10	To 01/28/11	01/31/11	To 02/04/11	(1575)	16	(788)	(2963)	(98)	37179	2	70	1.6	2.4
34	01/05/11	To 02/04/11	02/07/11	To 02/11/11	788	6	(388)	(763)	131	37967	2	60	2.2	2.4
35	01/12/11	To 02/11/11	02/14/11	To 02/18/11	375	21	(450)	(1425)	18	38342	4	70	1.8	2.4
36	01/19/11	To 02/18/11	02/21/11	To 02/25/11	13	5	(438)	(450)	3	38355	3	70	2	2.4
37	01/26/11	To 02/25/11	02/28/11	To 03/04/11	525	8	(1613)	(1613)	66	38880	2	60	2.4	2
38	02/02/11	To 03/04/11	03/07/11	To 03/11/11	2975	5	(575)	(788)	595	41855	3	50	2.4	2.4
39	02/09/11	To 03/11/11	03/14/11	To 03/18/11	(1850)	19	(763)	(2275)	(97)	40005	3	50	2.4	2.4
40	02/16/11	To 03/18/11	03/21/11	To 03/25/11	(2050)	14	(600)	(1075)	(146)	37955	3	60	0.8	2.4
41	02/23/11	To 03/25/11	03/28/11	To 04/01/11	(900)	10	(638)	(2400)	(90)	37055	2	60	2.2	2.4
42	03/02/11	To 04/01/11	04/04/11	To 04/08/11	3450	2	0	0	1725	40505	3	60	1.4	2.4
43	03/09/11	To 04/08/11	04/11/11	To 04/15/11	650	16	(613)	(800)	41	41155	3	60	1.4	2.4
44	03/16/11	To 04/15/11	04/18/11	To 04/22/11	2950	8	(1038)	(1325)	369	44105	2	70	2.4	2.4
45	03/23/11	To 04/22/11	04/25/11	To 04/29/11	1650	20	(413)	(963)	83	45755	2	70	0.2	2.2
46	03/30/11	To 04/29/11	05/02/11	To 05/06/11	813	50	(750)	(3275)	16	46568	3	60	1.8	2.2
47	04/06/11	To 05/06/11	05/09/11	To 05/13/11	3288	32	(1263)	(3488)	103	49856	3	60	2.4	2
48	04/13/11	To 05/13/11	05/16/11	To 05/20/11	0	0				49856				
49	04/20/11	To 05/20/11	05/23/11	To 05/27/11	0	0				49856				
50	04/27/11	To 05/27/11	05/30/11	To 06/03/11	0	0				49856				
51	05/04/11	To 06/03/11	06/06/11	To 06/10/11	0	0				49856				
52	05/11/11	To 06/10/11	06/13/11	To 06/17/11	1513	76	(1900)	(3888)	20	51369	4	60	0.4	2.2
53	05/18/11	To 06/17/11	06/20/11	To 06/24/11	(1800)	64	(1538)	(4300)	(28)	49569	4	50	2.2	1.8
54	05/25/11	To 06/24/11	06/27/11	To 07/01/11	(613)	54	(913)	(2863)	(11)	48956	4	50	1	2.2
55	06/01/11	To 07/01/11	07/04/11	To 07/08/11	725	38	(1038)	(2788)	19	49681	4	40	2.4	2.4
56	06/08/11	To 07/08/11	07/11/11	To 07/15/11	0	0				49681				
57	06/15/11	To 07/15/11	07/18/11	To 07/22/11	0	0				49681				
58	06/22/11	To 07/22/11	07/25/11	To 07/29/11	0	0				49681				
59	06/29/11	To 07/29/11	08/01/11	To 08/05/11	0	0				49681				
60	07/06/11	To 08/05/11	08/08/11	To 08/12/11	0	0				49681				
61	07/13/11	To 08/12/11	08/15/11	To 08/19/11	0	0				49681				
62	07/20/11	To 08/19/11	08/22/11	To 08/26/11	0	0				49681				
63	07/27/11	To 08/26/11	08/29/11	To 09/02/11	0	0				49681				
64	08/03/11	To 09/02/11	09/05/11	To 09/09/11	0	0				49681				
65	08/10/11	To 09/09/11	09/12/11	To 09/16/11	0	0				49681				
66	08/17/11	To 09/16/11	09/19/11	To 09/23/11	0	0				49681				
67	08/24/11	To 09/23/11	09/26/11	To 09/30/11	0	0				49681				
68	08/31/11	To 09/30/11	10/03/11	To 10/07/11	0	0				49681				
69	09/07/11	To 10/07/11	10/10/11	To 10/14/11	(4813)	22	(1738)	(5463)	(219)	44868	2	50	2.4	1.8
70	09/14/11	To 10/14/11	10/17/11	To 10/21/11	7238	24	(538)	(1025)	302	52106	2	50	2.4	1.8
71	09/21/11	To 10/21/11	10/24/11	To 10/28/11	(963)	32	(1463)	(2925)	(30)	51143	2	50	2.4	1.2
72	09/28/11	To 10/28/11	10/31/11	To 11/04/11	2775	60	(1125)	(2800)	46	53918	2	50	2.4	0.6
73	10/05/11	To 11/04/11	11/07/11	To 11/11/11	(3100)	40	(863)	(3750)	(78)	50818	2	50	2.4	0.6
74	10/12/11	To 11/11/11	11/14/11	To 11/18/11	0	0				50818				
75	10/19/11	To 11/18/11	11/21/11	To 11/25/11	0	0				50818				
76	10/26/11	To 11/25/11	11/28/11	To 12/02/11	0	0				50818				
77	11/02/11	To 12/02/11	12/05/11	To 12/09/11	1288	10	(488)	(1113)	129	52106	2	70	1.8	2.4

Week#	In-Sample Dates		Out-Of-Sample Dates		Osnp	OnT	ollt	odd	avosnp	Equity	Pw	N	aup	adn
78	11/09/11	To 12/09/11	12/12/11	To 12/16/11	450	10	(1225)	(1750)	45	52556	2	70	2.4	2
79	11/16/11	To 12/16/11	12/19/11	To 12/23/11	(363)	8	(1350)	(1575)	(45)	52193	2	50	2.4	2.4
80	11/23/11	To 12/23/11	12/26/11	To 12/30/11	1450	3	(338)	(338)	483	53643	2	50	2.4	2.4
81	11/30/11	To 12/30/11	01/02/12	To 01/06/12	1450	8	(850)	(1063)	181	55093	2	50	2.2	2.4
82	12/07/11	To 01/06/12	01/09/12	To 01/13/12	2950	10	(538)	(538)	295	58043	2	50	2.2	2.4
83	12/14/11	To 01/13/12	01/16/12	To 01/20/12	1788	15	(738)	(900)	119	59831	2	50	2	2.4
84	12/21/11	To 01/20/12	01/23/12	To 01/27/12	5163	15	(638)	(638)	344	64994	2	50	2.2	2.4
85	12/28/11	To 01/27/12	01/30/12	To 02/03/12	975	16	(775)	(2250)	61	65969	2	50	1.8	2.4
86	01/04/12	To 02/03/12	02/06/12	To 02/10/12	(50)	17	(1325)	(2625)	(3)	65919	3	70	2.2	1.8
87	01/11/12	To 02/10/12	02/13/12	To 02/17/12	725	10	(1275)	(1275)	73	66644	2	50	2.4	2.4
88	01/18/12	To 02/17/12	02/20/12	To 02/24/12	3400	4	(413)	(413)	850	70044	2	50	2.4	2.4
89	01/25/12	To 02/24/12	02/27/12	To 03/02/12	(1838)	12	(375)	(375)	(153)	68206	4	60	0.2	2.4
90	02/01/12	To 03/02/12	03/05/12	To 03/09/12	513	8	(475)	(963)	64	68719	2	50	2.4	1.8
91	02/08/12	To 03/09/12	03/12/12	To 03/16/12	300	10	(1125)	(2325)	30	69019	2	50	2.4	1.8
92	02/15/12	To 03/16/12	03/19/12	To 03/23/12	(1813)	7	(1713)	(2463)	(259)	67206	3	70	2.2	2.2
93	02/22/12	To 03/23/12	03/26/12	To 03/30/12	1600	4	(100)	(163)	400	68806	3	70	2.4	1.8
94	02/29/12	To 03/30/12	04/02/12	To 04/06/12	(338)	7	(975)	(1025)	(48)	68468	3	60	2.2	2
95	03/07/12	To 04/06/12	04/09/12	To 04/13/12	1750	6	(488)	(488)	292	70218	3	60	2.2	2.4
96	03/14/12	To 04/13/12	04/16/12	To 04/20/12	2188	5	(275)	(413)	438	72406	3	70	2.4	2.4
97	03/21/12	To 04/20/12	04/23/12	To 04/27/12	1100	5	(950)	(1500)	220	73506	3	60	2.4	2.4
98	03/28/12	To 04/27/12	04/30/12	To 05/04/12	513	5	(388)	(388)	103	74019	3	70	2.4	2.4
99	04/04/12	To 05/04/12	05/07/12	To 05/11/12	50	5	(888)	(1438)	10	74069	3	70	2	2.2
100	04/11/12	To 05/11/12	05/14/12	To 05/18/12	125	10	(1513)	(1513)	13	74194	3	70	2	2.2
101	04/18/12	To 05/18/12	05/21/12	To 05/25/12	3300	10	(625)	(725)	330	77494	2	70	2.4	1.4
102	04/25/12	To 05/25/12	05/28/12	To 06/01/12	(3313)	19	(838)	(3400)	(174)	74181	4	70	2.4	1.8
103	05/02/12	To 06/01/12	06/04/12	To 06/08/12	(1363)	7	(1550)	(2650)	(195)	72818	3	60	2.4	1.8
104	05/09/12	To 06/08/12	06/11/12	To 06/15/12	1663	9	(1300)	(1888)	185	74481	2	40	2.2	2.4
105	05/16/12	To 06/15/12	06/18/12	To 06/22/12	(2275)	14	(1700)	(3400)	(163)	72206	2	40	2.2	2.4

**Figure 1 Graph of Net Equity Applying the Walk Forward Filter Each Week
On EC 1min Bar Prices 6/25/2010 – 6/22/2012 Nth Order Polynomial Acceleration System**

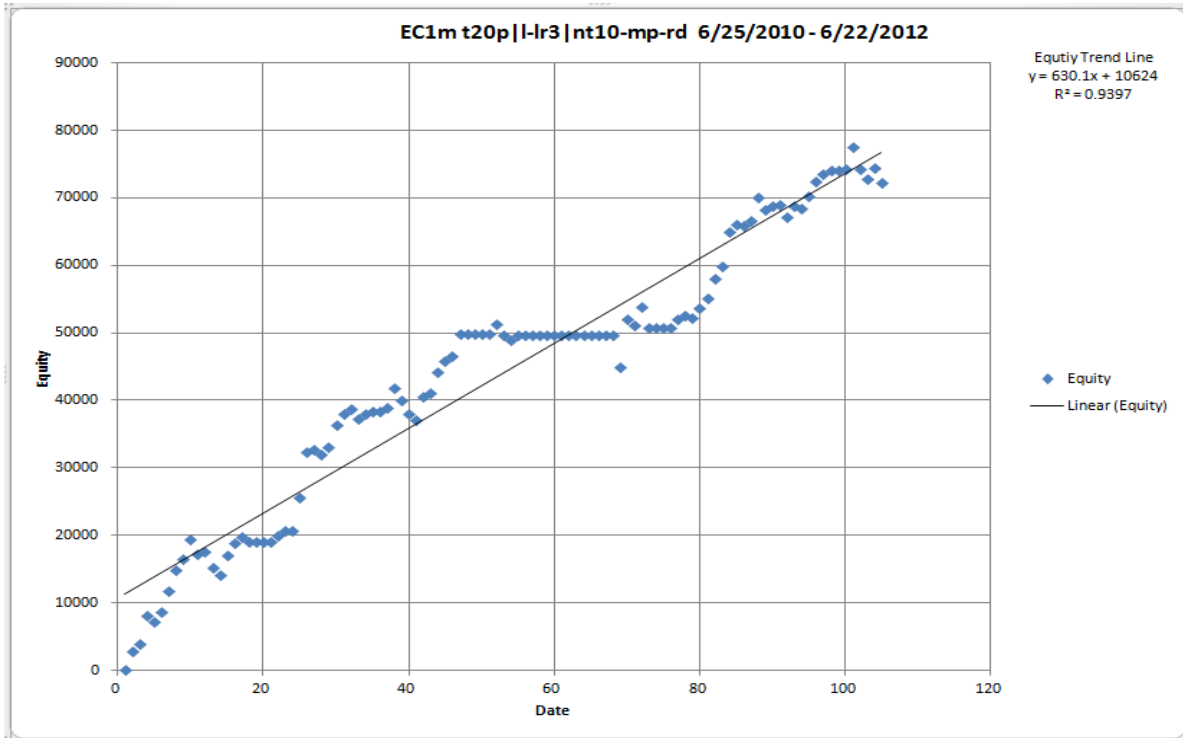


Figure 2 EC Nth Order Polynomial Acceleration 1 minute bar chart from 6/4/12-6/5/2012 example

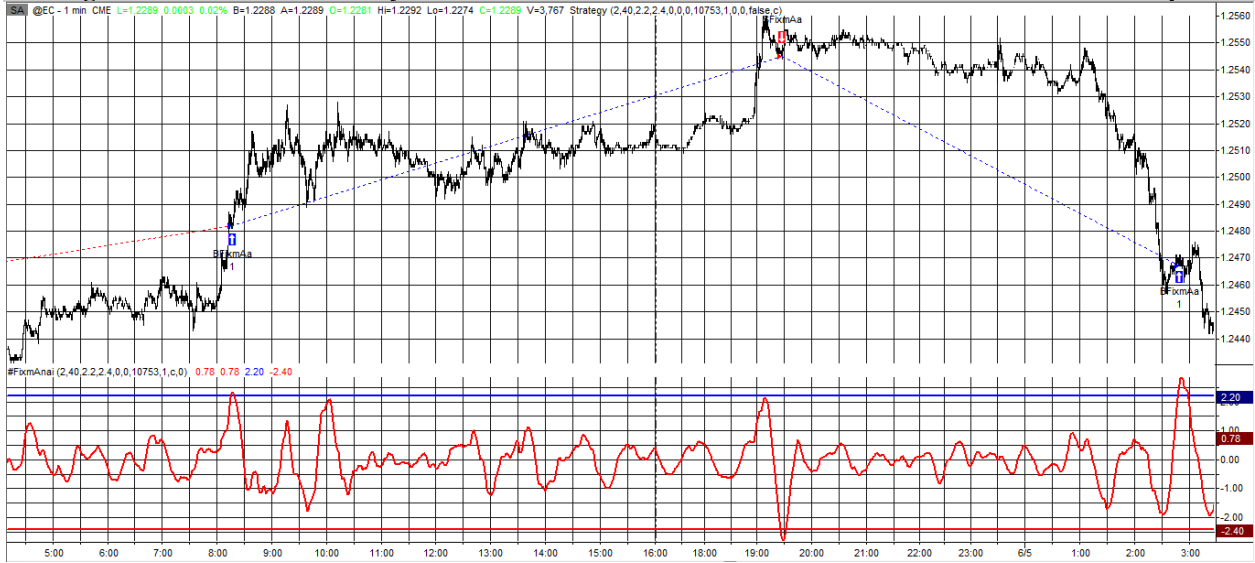


Figure 3 Partial output of the Walk Forward Metric Performance Explorer (WFMPE) EC-Mini 1 min bars Nth Order Polynomial Acceleration System

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	EC1FixmAn	20100625	20120622	#105	tnp>0						a[54798]	s26945	f15376							c=\$0	
2	Filter-Metric	tOnp	mOnp	aOnp	aOTr	aOnT	B0	%P	t	std	llw	eqDD	lr	#	eqTrn	eqR2	Blw	BE	ndd	tOnpNet	Prob
3	t20p l-lr3 nt10-mp-rd	72206	806	903	54.3	16.6	(14.6)	0.71	3.72	2171.1	-4813	-6501	3	80	630	94	17	23.1	9	72206	1.22E-06
4	t20p l-lr3-mp-rd	72206	806	903	54.3	16.6	(14.6)	0.71	3.72	2171.1	-4813	-6501	3	80	630	94	17	23.1	9	72206	1.22E-06
5	t20p l-lr3-b2	67805	794	848	49	17.3	(5.1)	0.7	3.19	2375.1	-4963	-7688	3	80	581	89	17	31.4	11.3	67805	2.68E-06
6	t20p l-lr3 nt10-b2	67805	794	848	49	17.3	(5.1)	0.7	3.19	2375.1	-4963	-7688	3	80	581	89	17	31.4	11.3	67805	2.68E-06
7	t20p l-lr3 nt10-mtnp	66728	788	834	49	17	(26.5)	0.66	3.33	2238.9	-4813	-8764	4	80	672	89	17	28.8	13.1	66728	3.24E-06
8	t20p l-lr3-mtnp	66728	788	834	49	17	(26.5)	0.66	3.33	2238.9	-4813	-8764	4	80	672	89	17	28.8	13.1	66728	3.24E-06

The WFMPE Filter Output Columns are defined as follows:

Row 1 EC1FixmAn is the PWFO File Stub, File Start Date(6/25/10), File End Date(6/22/12), **Number of weeks=#105**, **tnp>0**=In-Sample total net profit >0, **a**=average of bootstrap random picks. **s**= standard deviation of bootstrap random picks. **f**=number of different filters examined. **c**= slippage and round trip trade cost(c=\$0).

Filter = The filter that was run. Row 3 filter is t20p/l-lr2|nt10-mp-rd

Metric = The PWFO performance metric . For this filter, t20p/l-lr2|nt10

Metric= **mp-rd** = The median of the final trade profit minus the maximum rundown of the trade in the in-sample/test section

The is t20p/l-lr2|nt10-mp-rd filter produced the following average 105 week statistics on row 3.

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

mOsp = median oos net profit for the 105 weeks

aOsp = Average oos net profit for the 105 weeks

aanp = Average oos profit per trade

aOnT = Average number of oos trades per week

B0 = The 105 week trend of the out-of-sample weekly profits

%P = The percentage of oos weeks that were profitable

t = The student t statistic for the 105 weekly oos profits. The higher the t statistic the higher the probability that this result was not due to pure chance

std = The standard deviation of the 105 weekly oos profits

llw = The largest losing oos week

eqDD = The oos equity drawdown

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

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

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

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

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

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

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

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

Appendix A: an example of a truncated PWFO file with Out-Of-Sample end date of 6/22/12:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	1120517	1120615	1120622	EC1FixmAn																			
2	Pw	N	vup	vdn	mtnp	tnp	nT	pP	PF	std	t	mlb	lbr	mwb	wbr	awl	wlb	mru-p	mp-rd	wr	lr	mpft	mlos
3	2	30	0.2	0.2	-12.5	-4725	1559	47	0.93	122	-0.984	14	12697	22	15545	1.571	1.224	87.5	37.5	9	10	75	-62.5
4	2	40	0.2	0.2	-12.5	1625	1181	48	1.03	137	0.346	18	11966	29	16468	1.611	1.376	100	50	10	8	75	-62.5
5	3	40	0.2	0.2	0	-175	1703	51	1	115	-0.037	14	12659	19	16157	1.357	1.276	75	50	9	11	62.5	-50
6	2	50	0.2	0.2	-12.5	3225	933	48	1.07	151	0.699	23	12166	35	16592	1.522	1.364	112.5	50	7	9	75	-75
7	3	50	0.2	0.2	-12.5	100	1383	49	1	126	0.021	17	13268	23	15379	1.353	1.159	87.5	50	8	9	75	-62.5
8	2	60	0.2	0.2	-12.5	625	790	45	1.01	172	0.129	28	13143	41	15460	1.464	1.176	125	62.5	7	8	87.5	-87.5
9	3	60	0.2	0.2	0	5525	1128	51	1.11	138	1.194	21	12573	28	15982	1.333	1.271	87.5	50	10	8	75	-62.5
10	4	60	0.2	0.2	-12.5	-9300	1553	50	0.87	122	-1.94	16	13538	20	15080	1.25	1.114	75	50	9	9	62.5	-62.5
11	2	70	0.2	0.2	-25	-1125	692	45	0.97	181	-0.236	31	12777	48	15182	1.548	1.188	125	62.5	8	8	87.5	-87.5
12	3	70	0.2	0.2	-12.5	3900	963	50	1.08	149	0.841	23	12629	33	15740	1.435	1.246	100	62.5	10	9	87.5	-62.5
13	4	70	0.2	0.2	0	4825	1318	52	1.09	129	1.034	19	12849	24	15535	1.263	1.209	87.5	50	8	7	75	-62.5
14	2	30	0.4	0.2	-12.5	-5825	1383	47	0.91	128	-1.227	14	12746	23	15356	1.643	1.205	87.5	37.5	9	11	75	-62.5
15	2	40	0.4	0.2	-12.5	75	1053	48	1	145	0.016	19	12631	30	15861	1.579	1.256	100	50	11	8	75	-75
16	3	40	0.4	0.2	0	-200	1524	51	1	122	-0.042	14	12461	19	16159	1.357	1.297	75	50	9	11	62.5	-62.5
17	2	50	0.4	0.2	-12.5	3250	834	48	1.07	162	0.694	24	12254	36	16293	1.5	1.33	125	50	7	8	87.5	-75
18	3	50	0.4	0.2	0	3250	1209	50	1.06	135	0.691	17	13244	24	15207	1.412	1.148	87.5	50	9	7	75	-62.5

W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
In-Sample Section														Out-Of-Sample Section				
mlos	p/l	dd	llt	b0	r2	dev	b1	dy	b2	r22	mKr	e-3	eq10	osnp	odd	ollt	ont	aonp
-62.5	1.2	-6688	-1738	-3.2	-75.4	526.4	-2.79	-3.6	0	-75.6	-0.6	-212.5	-5.1	1025	-2975	-363	362	2.8
-62.5	1.2	-5438	-688	-0.2	-0.2	1037.7	-5.97	7.7	0.01	-37.7	-0.02	-200	1.1	5275	-1125	-388	269	19.6
-50	1.25	-5100	-1750	-0.2	-1	726.7	-4.61	4.4	0	-39.6	-0.03	-212.5	-0.1	-4175	-4525	-1200	413	-10.1
-75	1	-4588	-700	3.3	34.5	900.7	-1.17	6.4	0	30.1	0.37	-50	2.5	3625	-1638	-338	225	16.1
-62.5	1.2	-5250	-625	-0.4	-1.8	984.5	-6.05	6.4	0	-51.6	-0.04	-75	0.3	2500	-2013	-413	316	7.9
-87.5	1	-4613	-538	1	5.3	802.1	0.35	0.8	0	3.7	0.13	400	0.5	-125	-2625	-1113	192	-0.7
-62.5	1.2	-4850	-700	3.6	31	1473.6	-6.4	15.7	0.01	64.9	0.25	-250	5.4	3100	-1425	-325	267	11.6
-62.5	1	-10538	-1738	-5.4	-82	832.2	-5.7	-5.7	0	-81.7	-0.65	-162.5	-8.9	-450	-2550	-363	346	-1.3
-87.5	1	-3638	-463	1.9	16.6	583	-11.47	12.4	0.02	-2.8	0.32	200	0.5	500	-1525	-1113	156	3.2
-62.5	1.4	-4138	-688	1.2	10.3	707.8	4.12	-1	0	8	0.17	-37.5	1.5	2650	-2250	-375	235	11.3
-62.5	1.2	-4063	-538	3.7	46.8	1255.1	-6.28	14.5	0.01	81.5	0.29	-150	5.5	850	-1250	-413	310	2.7
-62.5	1.2	-6688	-1738	-3.8	-74.4	755.7	-1.24	-7.3	0	-81.3	-0.5	-175	-6	1825	-2125	-350	318	5.7
-75	1	-4838	-688	-1.1	-8.7	757.4	-4.88	4.1	0	-29.3	-0.14	-225	-0.4	3650	-1550	-375	253	14.4
-62.5	1	-5625	-1750	-0.2	-0.7	617.4	-5.09	5.1	0	-49.9	-0.03	-200	0.1	-4875	-5000	-1200	365	-13.4
-75	1.167	-3363	-700	3.8	40.4	848	7.06	-1.7	-0.01	48.3	0.45	-75	2.2	600	-2425	-1025	190	3.2
-62.5	1.2	-4850	-700	2.3	33.1	851.2	-0.89	7.2	0	47.7	0.27	-300	3.9	1625	-2163	-413	300	5.4

Appendix B: 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 $b_0 + b_1 * t + b_2 * t^2 + b_3 * t^3 + \dots + b_n * t^n$ 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.

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

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

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 *acceleration* by the equations:

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

$$\text{Acceleration}(t+1) = d^2P_f / d^2t = 2c_0 + 6d_0 * (T+1) + 12e_0 * (T+1)^2$$

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

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