

Trading QQQ ETF 5min Bars Using the nth Order Fixed Memory Polynomial Velocity Algorithm
Walk Forward in-sample 10 Trading weekdays and out-of-sample 1 Trading weekday.
1/3/2022 to 8/4/2023
Working Paper August2023 Copyright © 2023 Dennis Meyers

Disclaimer

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

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

The velocity strategy that we will use here to trade the Invesco QQQ Trust Series ETF (**QQQ**) is called the nth Order Polynomial Velocity Strategy. The nth Order Adaptive Polynomial Velocity Strategy has several unknown inputs that we must determine before we can use this strategy to trade. These unknown inputs are the polynomial order or degree, the optimum number of lookback prices we need to determine the coefficients of the polynomial and finally the velocity thresholds. Here we will use Walk Forward Optimization and out-of-sample testing to determine the “best” polynomial inputs as well as how these inputs change over time. We will use the nth Order Fixed Memory Polynomial Velocity Strategy to trade the QQQ ETF on an intraday basis using 5-min bar price data from 01/03/2022 to 8/4/2023.

The nth Order Fixed Memory Velocity Strategy Defined

The least squares forecast nth 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 nth Order Fixed Memory Polynomial **Velocity(T+1)** is constructed from the equation shown in the “Math” appendix.

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 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 nth 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 nth order polynomial velocity, the number of lookback prices, **N**, needed to compute the polynomial coefficients and the threshold amounts **vup** and **vdn**. When the nth order degree velocity is greater than the threshold amount **vup** we will go long. When the velocity is less than the threshold amount **-vdn** we will go short.

Buy Rule:

IF Velocity is greater or equal than the threshold amount **vup and Velocity[1] is less than vup** then buy at the market.

Sell Rule:

IF Velocity is less than or equal than the

threshold amount **-v_{dn} and Velocity[1] is greater than -v_{dn}** then sell at the market.

Where Velocity[1] is the velocity on the previous bar.

Intraday Bars Exit Rule:

Close the position at 1555 EST (No trades will be carried overnight).

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. **v_{up}**, the threshold amount that velocity must be greater than to issue a buy signal.
4. **v_{dn}**, the threshold amount that velocity must be less than to issue a sell signal.

As mentioned, to test this Strategy we will use five-minute bar prices of the QQQ ETF traded on the NYSE and known by the symbol QQQ for the 389 trading days from January 3, 2022 to August 4, 2023.

We will test the FixmVn strategy with the above QQQ ETF 5 min bars on a **walk forward basis**, where the in-sample (**IS**) will be 10 trading weekdays and the out-of-sample (**OOS**) will be the next trading weekday following as will be described below.

What Is a Walk Forward Optimization with In-Sample Section and Out-Of-Sample Sections?

Whenever we do a TradeStation or Multicharts (TS/MC) optimization on many different strategy inputs, TS/MC generates a report of performance metrics (total net profits, number of losing trades, etc.) vs these different strategy inputs. If the report is sorted on say the total net profits (**tnp**) performance metric column, then the highest **tnp** would correspond to a certain set of inputs. This is called an **in-sample (IS) 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" it is important to see how the strategy inputs chosen from the in-sample section perform on out-of-sample data.

What do we mean by "**curve fitting**" or **data mining**? As a simple example, suppose you were taking a subway to work. In the subway car you are in, suppose you counted the number of blond women in that car and suppose the percent of blond women vs all other women hair colors was 80%. Being that you cannot observe what is in the other subway cars, you would assume that all the other subway cars and perhaps all women had the same percentage of blond hair. This observation was due to chance. That is an example of curve fitting. The same goes for combinatorial searches. You are observing results from a finite sample of data without knowing the data outside the sample you examined.

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 performance metric selection procedure, which we will call a **filter**, used to select the strategy input parameters from the optimization run. For instance, a **filter** example 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 or losses for each of the out-of-sample sections. Using this method, we now have "x" number of independent out-of-sample section profit and losses from our **filter**. If we take the average of these out-of-sample section net profits and losses, then we

will have an estimate of how our strategy will perform on average. Due to the Central Limit Theorem, as your 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 or some other performance metric? Surely the price noise cancels itself out with such a large number of in-sample prices and trades. Unfortunately, nothing could be farther from the truth! Optimization is a misnomer and should really be called combinatorial search. As stated above, whenever we run a combinatorial search over many different combinations of input parameters on noisy data on a fixed number of prices, **no matter how many**, the best performance parameters found are guaranteed to be due to **“curve fitting”** the noise and signal. 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 inputs parameter combinations, the best performance parameters will be from those strategy input variables that are able to produce profits from the price pattern **and** the random spurious movements. While the price patterns will repeat, the same spurious price movements will not. If the spurious price 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 intraday 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 or combinatorial searches, also called back testing, with no out-of-sample testing cause losses when traded in real time from something that looked great in the in-sample section.

To gain confidence that our input parameter selection method or filter, using the optimization output of the in-sample data, will produce profits, we must test the input parameters we found in the in-sample section on out-of-sample data. In addition, we must perform the in-sample/out-of-sample analysis many times. Why not just do the out-of-sample analysis once or just 10 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 out-of-sample profit “luck”. That is, by pure chance we may have chosen some input parameter set that did well in the in-sample section data **and** the out-of-sample section data. In order to minimize this type of “luck”, statistically, we must repeat the walk forward out-of-sample (**OOS**) analysis over many (>50) in-sample/out-of-sample sections and take an average over all out-of-sample sections. This average gives us an expected out-of-sample return and a standard deviation of out-of-sample returns which allows us to statistically estimate the expected equity and its range for N out-of-sample periods in the future.

Finding The FixmVn Strategy Parameters Using Walk Forward Optimization

There are four strategy parameters to find, pw , N , vup , vdn .

For the test data we will run the TS or MC optimization engine on **QQQ** 5 min price bars from 01/03/2022 to 8/4/2023 with the following optimization ranges for the FixmVn strategy inputs. This will create **394, 10 weekday in-sample periods each followed by a 1 day out-of-sample period** (See Figure 1 for the in-sample/out-of-sample periods). The days are weekdays only. Weekdays where the OOS falls on an exchange holiday or partial days are eliminated. Holidays that fall on a weekday create a 9-day **IS**. All other **IS** periods consist of 10 trading weekdays. The optimization ranges are:

1. **pw=degree from 1 to 3**
2. **N from 5 to 20 in steps of 1.**
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 = 13.8, iNorm=1 (See Appendix 3, the Normalization Multiplier)**

The above pw , n , vup , vdn will produce 9408 different input combinations or cases of the strategy input parameters for each of the 389 in-sample/out-of-sample files for the 19 months of 5 min bar QQQ data. Note the first 10-day trading in-sample period starts at 01/18/2022.

Finding the Best in-sample Metric Filter.

The question we are attempting to answer statistically is which performance metric or combination of performance metrics (which we will call a *filter*) applied to the in-sample section will produce in-sample strategy inputs that produce statistically valid average 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 section that can give us strategy inputs that will produce, on average, good trading results in the future.

When TradeStation(TS) or MultiCharts(MC) does an optimization over many combinations of inputs, it creates an output page that has as its rows each strategy input combination and as its columns various trading performance measures such as Profit Factor, Total Net Profits, etc. An example of a simple filter would be to choose the strategy input optimization row in the in-sample section that had the highest Net Profit or perhaps a row that had the best Profit Factor with their associated strategy inputs. Unfortunately, it was found that this type of simple metric performance 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 in-sample selection of strategy inputs.

The combination metric filters are found by a program called WFME64v8x. Details of this program can be found at <https://meyersanalytics.com/wfme.html>.

All PWFO file metrics used by the WFME64v8x are described at <https://meyersanalytics.com/Walk-Forward-Optimization.html>.

We will use the WFME64 v8x program to find one in-sample combination-metric filter applied to each in-sample section which gives a set of strategy inputs which are then applied to each following out-of-sample section. This will consist of 389 in-sample and out-of-sample sections from 1/18/2022 to 8/4/2023.

Here is a metric combination *filter* found by the WFME64 v8x program that was used in this paper. High profit factors (**pf**) in the in-sample section usually mean poor performance in the out-of-sample-section. This is a kind of reversion to the mean. So, in the in-sample(*IS*) section we eliminate all strategy input rows that have a **pf**>**3.5**. We also wish to limit the number of losing trades in a row (**lr**) in the *IS* period to 3 or less (**lr**≤**3**). In addition, we want the R2 equity trend line correction to be <80, **r2**<**80**. Using the **pf-lr-r2** elimination screen, as described, there can still be 100's of rows left in the in-sample section. The PWFO generates the performance metric named **eq2V**. This metric, **eq2V**, is the **Velocity of the fitted IS Equity by a 2nd Order Polynomial Line**. Let us choose the **50** rows in the in-sample section that contain the **maximum eq2V** values from the rows that are left from the **pf-lr-r2** screen. In other words, we sort **eq2V** from high to low, eliminate the rows that have **lr**>**3**, **pf**>**3.5**, **r2**>**80** and then choose the largest **top 50** rows of whatever is left. This filter will now leave 50 cases or rows in the in-sample section that satisfy the above filter conditions. We call this filter **t50eq2V | p≤3.5lr≤3r2≤80** where **t50eq2V** means the top or maximum **50 eq2V** rows left *after* the **pf-lr-r2** in-sample row elimination. Suppose for this filter, within the 50 in-sample rows that are left, we want the row that has the largest value of the metric called **aWb**. **aWb** is the average of the bars in the winning trades.

We abbreviate this final filter as **t50eq2V | p≤3.5lr≤3r2≤80-aWb**. For each in-sample section this filter leaves only one row in the in-sample section with its associated strategy inputs and following out-of-sample net profit in the out-of-sample section using the strategy inputs found in the in-sample section. This **t50eq2V | p≤3.5lr≤3r2≤80-aWb** filter is then applied to each of the 389 in-sample sections which give 389 sets of strategy inputs that are used to produce the corresponding 389 out-of-sample performance results. The average out-of-sample performance is calculated from these 389 out-of-sample performance results. In addition, many other important out-of-sample performance statistics for this filter are calculated and summarized.

Figure 2 shows such a computer run along with a small sample of other WFME64 filter combinations that are constructed in a similar manner. **Row 3** of the sample output in **Figure 2** shows the results of the filter discussed above.

From this run, we chose the filter on row 3 of the Figure below. That is,

t50eq2V | p<3.5| r<3r2<80-aWb.

	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	
1	QQQ5mFixmV10x1dxa	s01/18/22	e08/04/23	#389	AnyTnp																						
2	Filter-Metric	toGP	toNP	aoGP	aoTr	ao#T	#	std	skew	kur	t	oW oL	%Wtr	%P	LLp	eqDD	wpr	lpr	v20	acc	c=\$4	KTau	eqR2	Blw	BE	tkr bl	Prob
3	t50eq2V pf<3.5 r<3r2<80-aWb	28594	27454	131	100.3	1.3	219	374	0.211	3.02	5.17	1.35	61	64	-772	-1601	8	4	68	-0.35	96	89	29	61	2486	1.63E-12	
4	t50wr pf<2 r2<80-eqTrn	31403	27279	87	30.5	2.8	363	396	0.418	3.62	4.16	1.5	49	58	-915	-2909	11	8	59	-0.51	81	83	102	94	293	1.90E-05	
5	t10ktau pf<5 r<5r2<80-eq2A	28427	27063	138	83.4	1.7	206	406	-0.158	3.97	4.88	1.79	53	64	-1387	-2621	7	6	43	-0.53	88	85	99	69	534	1.37E-13	
6	t10ktau pf<2 r2<80-eq2A	28341	26929	136	80.3	1.7	208	408	-0.14	3.87	4.82	1.79	52	63	-1387	-2621	7	6	42	-0.51	88	85	99	70	518	3.04E-13	
7	t50eqTrn pf<3 r<3-aWb aLb	27552	26512	144	106	1.4	192	390	0.161	2.98	5.1	1.46	60	66	-765	-1831	12	3	27	-0.45	95	85	41	63	1597	7.94E-15	
8	t50eq2V pf<4.5 r<3r2<80-aWb	27303	26163	123	95.8	1.3	222	376	0.212	3.01	4.88	1.38	60	63	-772	-1902	9	4	66	-0.35	94	86	56	69	1021	3.08E-11	
9	t50eq2V pf<5 r<3r2<80-aWb	27157	26029	123	96.3	1.3	221	380	0.155	3.06	4.8	1.31	61	63	-849	-2477	9	4	79	-0.32	93	84	65	71	810	3.16E-11	
10	t20tWb tLb pf<2 r<3-eq2V	27366	25938	124	76.7	1.6	220	429	0.074	4.5	4.3	1.52	55	62	-1557	-1879	8	3	31	-0.4	90	87	52	88	732	3.02E-11	
11	t50eq2V pf<4 r<3r2<80-aWb	26890	25754	122	94.7	1.3	220	381	0.167	2.97	4.75	1.32	61	63	-772	-2364	8	5	68	-0.33	93	87	58	72	918	4.14E-11	
12	t50eq2V pf<5 r<3r2<80-aWb	26617	25481	122	93.7	1.3	218	386	0.078	3.11	4.66	1.27	61	63	-956	-1882	9	4	66	-0.27	95	91	39	75	1382	4.43E-11	
13	t10ktau pf<2 r<3r2<80-eq2A	26403	25243	131	91	1.4	202	404	-0.113	4.09	4.6	1.69	54	61	-1463	-2656	10	6	43	-0.55	84	83	100	77	415	1.91E-12	
14	t50eqTrn pf<3 aWb aLb	25697	24645	134	97.7	1.4	192	391	0.192	3	4.75	1.48	58	65	-765	-1831	12	3	27	-0.45	95	84	41	73	1276	4.81E-13	
15	t50aWT LT pf<2 r<3-aLTr	27494	24626	96	38.3	2.5	285	399	-0.484	8.23	4.09	1.48	50	60	-2348	-2348	8	4	70	-0.1	91	93	52	98	675	9.82E-07	

Bootstrap Probability of Filter Results.

Using modern "Bootstrap" techniques, we can calculate the probability of obtaining our filter's total out-of-sample *net* profits by chance. Here is how the bootstrap technique is applied. Suppose as an example, we have 500 files of in-sample/out-of-sample data. A mirror random filter is created. Instead of picking an out-of-sample net profit (OSNP) from a filter row as before, the mirror filter picks a *random* row's OSNP in each of the 500 files. We repeat this random picking in each of the 500 files 5000 times. Each of the 5000 mirror filters will choose a random row's OSNP of their own in each of the 500 files. At the end, each of the 5000 mirror filters will have 500 *random* OSNP's picked from the rows of the 500 files. The sum of the 500 random OSNP picks for each mirror filter will generate a random total out-of-sample net profit (toNP) or final random equity. The average and standard deviation of the 5000-mirror filter's different random toNPs will allow us to calculate the chance probability of our above chosen filter's toNP. Thus, given the mirror filter's bootstrap random toNP average and standard deviation, we can calculate the probability of obtaining our chosen filter's toNP by pure chance alone. Figure 3 lists the 5000-mirror filter's bootstrap average for our 389 out-of-sample files of **\$2.5** with a bootstrap standard deviation of **\$17.6**. (Side Note. The average is the average per out-of-sample period. So, the average for the random selection would be the random toNP/389 and the average for the filter would be the filter toNP/# of OOS periods traded or 27454/219 = 125.4). The probability of obtaining our filters average daily net profit of **125.4** is **1.63 x 10⁻¹²** which is **6.96** standard deviations from the bootstrap average of 2.5. For our filter, in row 3 above, the expected number of cases that we could obtain by pure chance that would match or exceed **\$125.4** is $[1 - (1 - 1.63 \times 10^{-12})^{345960}] \sim 345960 * 1.63 \times 10^{-12} = 0.000005639$ where **345960** is the total number of different filters we looked at in this run. This number is much less than one, so the probability is small that our result was due to pure chance.

Results

Figure 1 presents a graph of the equity curve generated by using the filter on the 389 days from 1/18/2022 to 8/4/2023. The equity curves are plotted from Equity and Net Equity columns in Table 1. Plotted on the equity curves is the 2nd Order Polynomial curve. 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 QQQ Daily Closing prices superimposed on the Equity Chart.

Figure 2 presents a plot of the FixmVn Strategy buy/sells and the FixmVn Indicator on the QQQ 5min bars for 7/27/23 to 8/4/2023.

Table 1 below presents a table of the 389 in-sample and out-of-sample windows, the **Filter** selected in-sample strategy inputs and the daily out-of-sample profit/loss results using the filter described above.

Discussion of Strategy Performance

In Figure 3, Row 3 of the spreadsheet filter output are some statistics that are of interest for our filter. An interesting statistic is **Blw**. **Blw** is the maximum number of days the OSNP equity curve failed to make a new high. **Blw** is **29** days for this filter. This means that 29 trading days was the longest time that the equity for this strategy failed to make a new equity high. **%Wtr** is the percentage of all OOS trades that were wins or positive. For this filter, the **%Wtr=61%**. **%P** is the % winning oos days, **%P=64%**. The average oos winning trade to the average oos losing trade ratio(**oW|oL**) was **1.35**. **wpr=8** is the maximum number of consecutive winning oos periods(days) in a row and **lpr=4** is the maximum number of consecutive losing oos periods(days) in a row. The largest losing day was **(\$772)**. The Maximum drawdown in this period was **(\$1601)** which occurred from 5/5/22 to 5/12/22.

In Figure 1, which presents a graph of the equity curve using the filter on the 389 trading days of out-of-sample data, notice how the equity curve follows the 2nd order polynomial trend line with an R² of 0.965. The R² only dropped to 0.962 for the net equity curve.

Using this filter, the strategy was able to generate \$27454 net equity after commissions of \$0 (many brokers today, 2023, don't charge commissions) and slippage of \$4 trading 100 QQQ ETF shares for 389 days. This period of time from 1/18/2 to 8/4/23 was a volatile down then up market. Yet the FixmVn strategy was able to adapt quite well.

In observing Table 1 we can see that this strategy and filter made trades from a low of no trades/day to a high of 5 trades/day with an average of 1.3 trades/day on the days it traded. For the no trade days, the strategy **filter** in the in-sample section didn't satisfy the metric filter and no trades were made the next trading day. The **filter** traded 219 days out of the 389 days or about 56% of the time.

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 FixmVn Strategy Equity Applying the Walk Forward Filter Each Day on the in-sample section on QQQ 5min Bar Prices 1/18/2022 to 8/4/2023

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 QQQ Daily Closing prices superimposed on the Equity Chart.

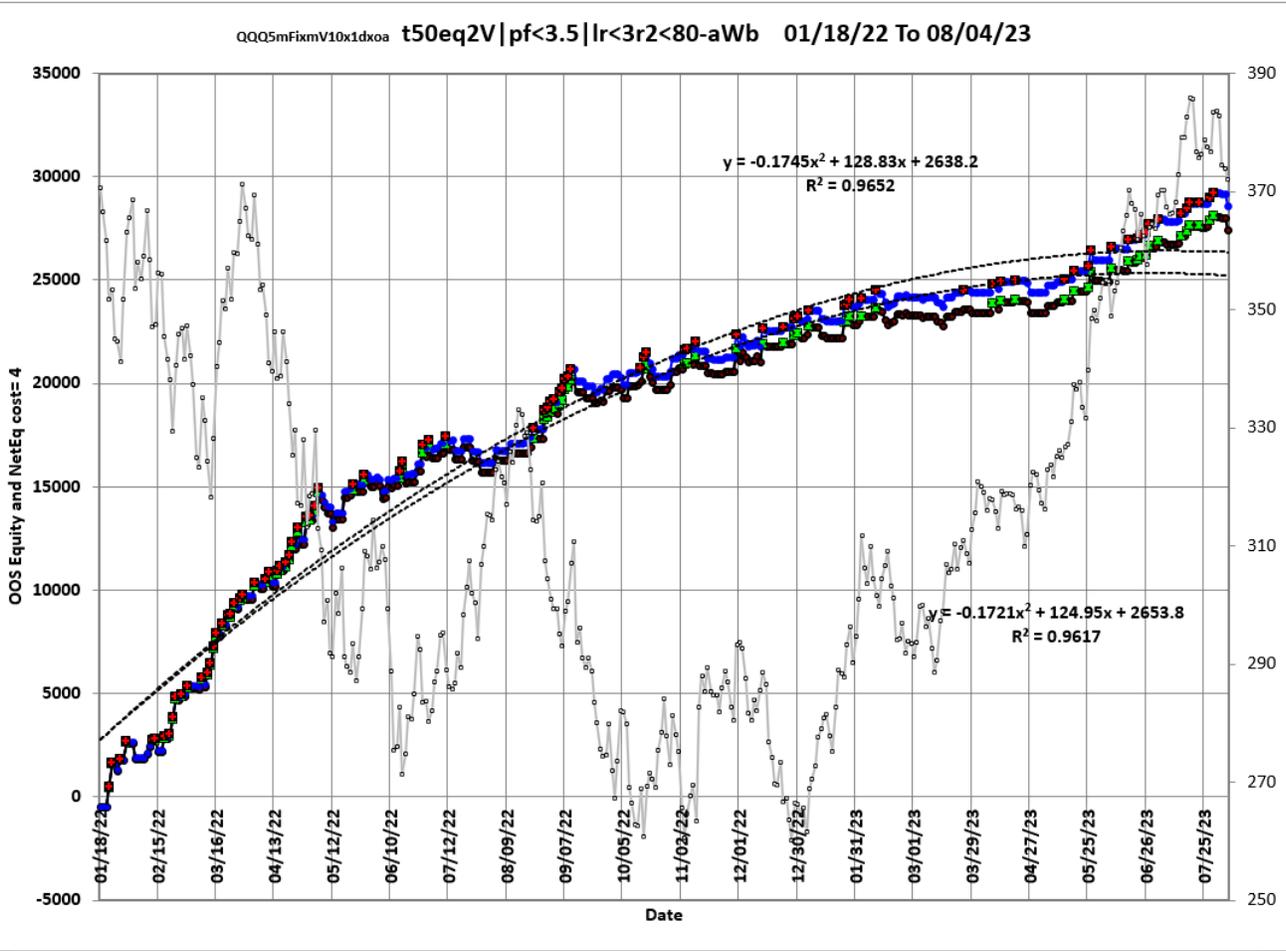


Figure 2 Walk Forward Out-Of-Sample Performance Summary for nth Order Fixed Memory Polynomial Velocity Strategy QQQ 5-minute bar chart from 7/27/23 to 8/4/2023

Note: The FixmV inputs change every day so we plotted the inputs that did best in the QQQ5mFixmVn20x1d.pdf workpaper at <https://meyersanalytics.com/publications2/QQQ5mFixmVn20x1d.pdf>.



Figure 3 Partial output of the Walk Forward Strategy Inputs with Metric Filters (WFME) QQQ ETF 5 min bars Using The FixmVn Strategy

The WFME Filter Output Columns are defined as follows: OOS=out-of-sample

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
Filter-Metric	toGP	toNP	aoGP	aoTr	ao#T	#	std	skew	kur	t	oW oL	%Wtr	%P	LLp	eqDD	wpr	lpr	v20	acc	KTau	eqR2	Blw	BE	tkr bl	Prob
QQQ5mFixmV10x1dxxoa	s01/18/22	e08/04/23	#389	AnyTnp							ISnt2		a2.5	s17.6					c-\$4					f345960	
t50eq2V pf<3.5 r<3r2<80-aWb	28594	27454	131	100.3	1.3	219	374	0.211	3.02	5.17	1.35	61	64	-772	-1601	8	4	68	-0.35	96	89	29	61	2486	1.63E-12
t50wr pf<2 r2<80-eqTrn	31403	27279	87	30.5	2.8	363	396	0.418	3.62	4.16	1.5	49	58	-915	-2909	11	8	59	-0.51	81	83	102	94	293	1.90E-05
t10ktau pf<2 r<5r2<80-eq2A	28427	27063	138	83.4	1.7	206	406	-0.158	3.97	4.88	1.79	53	64	-1387	-2621	7	6	43	-0.53	88	85	99	69	534	1.37E-13
t10ktau pf<2 r2<80-eq2A	28341	26929	136	80.3	1.7	208	408	-0.14	3.87	4.82	1.79	52	63	-1387	-2621	7	6	42	-0.51	88	85	99	70	518	3.04E-13
t50eqTrn pf<3 r<3-aWb aLb	27552	26512	144	106	1.4	192	390	0.161	2.98	5.1	1.46	60	66	-765	-1831	12	3	27	-0.45	95	85	41	63	1597	7.94E-15
t50eq2V pf<4.5 r<3r2<80-aWb	27303	26163	123	95.8	1.3	222	376	0.212	3.01	4.88	1.38	60	63	-772	-1902	9	4	66	-0.35	94	86	56	69	1021	3.08E-11
t50eq2V pf<5 r<3r2<80-aWb	27157	26029	123	96.3	1.3	221	380	0.155	3.06	4.8	1.31	61	63	-849	-2477	9	4	79	-0.32	93	84	65	71	810	3.16E-11
t20tWb tLb pf<2 r<3-eq2V	27366	25938	124	76.7	1.6	220	429	0.074	4.5	4.3	1.52	55	62	-1557	-1879	8	3	31	-0.4	90	87	52	88	732	3.02E-11
t50eq2V pf<4 r<3r2<80-aWb	26890	25754	122	94.7	1.3	220	381	0.167	2.97	4.75	1.32	61	63	-772	-2364	8	5	68	-0.33	93	87	58	72	918	4.14E-11
t50eq2V pf<3 r<3r2<80-aWb	26617	25481	122	93.7	1.3	218	386	0.078	3.11	4.66	1.27	61	63	-956	-1882	9	4	66	-0.27	95	91	39	75	1382	4.43E-11
t10ktau pf<2 r<3r2<80-eq2A	26403	25243	131	91	1.4	202	404	-0.113	4.09	4.6	1.69	54	61	-1463	-2656	10	6	43	-0.55	84	83	100	77	415	1.91E-12
t50eqTrn pf<3 aWb aLb	25697	24645	134	97.7	1.4	192	391	0.192	3	4.75	1.48	58	65	-765	-1831	12	3	27	-0.45	95	84	41	73	1276	4.81E-13
t50aWT LT pf<2 r<3-aLTr	27494	24626	96	38.3	2.5	285	399	-0.484	8.23	4.09	1.48	50	60	-2348	-2348	8	4	70	-0.1	91	93	52	98	675	9.82E-07

The WFME64 v8X AVE File Output Cols are defined as follows.

• **Row 1**

Column: A=The PWFO Stub, **B**=File Start Date, **C**=File End Date, **D**= Number of oos periods (in this example weeks), **N**= Bootstrap average, **O**= Bootstrap Standard Deviation, **AG**=Number of filters run, **U**= Cost and slippage per trade. **Z**=Number of filters examined.

• **Row 2 to Last Row Columns: A through AG**

Col A: The Strategy Input/Filter Names Example Row 3: **b50mLb|p<4lr<3r2<80:**

Col B: toGP - Total out-of-sample(oos) gross profit for these 512 oos periods (= weeks).

Col C: toNP - Total out-of-sample(oos) Net profit (toGP-Number of Trade Weeks*cost) for the 512 oos periods.

Col D: aoGP - Average oos gross profit for the 481 # oos periods

Col E: # - The number of oos periods this filter produced a profit or loss. Note for some oos periods there can be no strategy inputs that satisfy a given filters criteria and no trades will be made during that period.

Col F: aoTr - Average oos profit per trade

Col G: ao#T - Average number of oos trades per week

Col H: std - he standard deviation of the 512 oos period profits and losses

Col I: skew - The Skew statistic of the 512 oos period profits and losses

Col J: kur - he kurtosis statistic of the 512 oos period profits and losses

Col K: t - The student t statistic for the 512 oos periods. The higher the t statistic the higher the probability that this result was not due to pure chance.

Col L: oW|oL - Ratio of average oos winning trades divided by average oos losing trades.

Col M: %Wtr - he percentage if oos winning trades.

Col N: %P - percent of all oos periods that were profitable.

Col O: LLp - The largest losing oos period

Col P: eqDD - The oos equity drawdown

Col Q: wpr - The largest number of winning oos periods (weeks) in a row.

Col R: lpr - the largest number of losing oos periods in a row.

COLS: v20 - the equity velocity for the latest 20 periods

Col T: acc - The acceleration of a 2nd order polynomial fit to the oos equity curve.

Col U: $KTau^2$ - 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)

Col V: $eqR2$ - The correlation coefficient(R^2) of a straight-line fit to the equity curve.

Col W: Blw - The maximum number of oos periods the oos equity curve failed to make a new high.

Col X: BE - Break even in oos periods. Assuming the average and standard deviation are from a normal distribution, this is the number of oos periods you would have to trade to have a 98% probability that your oos equity is above zero.

Col Y: $t_{kr/bl}$ = $t * ktau * eqR2 / Blw$ a measure of how good the filter fit is.

Col Z: $Prob$ - The probability that the filters toNP was due to pure chance.

**Table 1 Walk Forward Out-Of-Sample Performance Summary for the
QQQ 5-min FixmVn Strategy**

QQQ-5 min bars 1/18/2022 - 8/4/2023.

QQQ 5 min bars 1/18/2022 - 8/4/23 OOS daily performance using the filter,
t50eq2V|pf≤3.5|lr≤3|r2≤80-aWb

on each in-sample segment. The input values *N*, *vup*, *vdn* are the values found from applying the filter to the in-sample (IS) section and used on the following out-of-sample section.

In-sample Section Filter: t50eq2V|pf≤3.5|lr≤3|r2≤80-aWb

Where:

osgp = Daily out-of-sample gross profit in \$

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

osnp\$4 = Daily out-of-sample Net Profit in \$ = **osgp-ont*4**

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

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

OSS Equity = Running Sum of daily out-of-sample gross profits \$

OSS NetEq = running sum of the daily out-of-sample net profits (osnp\$4) in \$

IS pw = the Fixed Memory Polynomial Velocity power exponent.

IS N = N the lookback period

IS vup, the threshold amount that velocity has to be greater than to issue a buy signal.

IS vdn, the threshold amount that velocity has to be less than to issue a sell signal.

Note: Zero rows indicate that no out-of-sample trades were made on that oss day.

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
01/18/22	2	8	1.75	3	(474)	(482)	2	-380	-474	(474)	(482)
01/19/22	1	13	3.25	3	0	0	0	0	0	(474)	(482)
01/20/22	1	15	2.75	3.25	0	0	0	0	0	(474)	(482)
01/21/22	2	17	3.5	0.25	935	927	2	0	0	461	445
01/24/22	1	18	2.75	3	1173	1165	2	0	0	1634	1610
01/25/22	1	10	3	2.75	(37)	(41)	1	-37	-37	1597	1569
01/26/22	1	7	3.25	3	(265)	(273)	2	-382	-382	1332	1296
01/27/22	1	5	3	3	481	473	2	-61	-61	1813	1769
01/28/22	2	17	3.25	3.5	0	0	0	0	0	1813	1769
01/31/22	2	17	3.25	3.5	850	846	1	0	0	2663	2615
02/01/22	2	17	3.25	3.5	0	0	0	0	0	2663	2615
02/02/22	2	17	3.25	3.5	0	0	0	0	0	2663	2615
02/03/22	2	20	2	3.25	(772)	(780)	2	-776	-776	1891	1835
02/04/22	1	20	3.5	2.75	0	0	0	0	0	1891	1835
02/07/22	1	20	3.5	3.5	0	0	0	0	0	1891	1835

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
02/08/22	1	20	3.25	3.5	0	0	0	0	0	1891	1835
02/09/22	1	9	3	3.5	270	266	1	0	0	2161	2101
02/10/22	1	7	3.5	2	308	304	1	0	0	2469	2405
02/11/22	1	6	3	2.5	263	259	1	0	0	2732	2664
02/14/22	1	15	2.25	0.75	56	52	1	0	0	2788	2716
02/15/22	1	13	2.5	1.25	(529)	(537)	2	-396	-529	2259	2179
02/16/22	2	15	2.75	2.75	0	0	0	0	0	2259	2179
02/17/22	2	15	2.75	2.75	654	650	1	0	0	2913	2829
02/18/22	3	19	3.5	2	0	0	0	0	0	2913	2829
02/22/22	3	19	3.5	2	110	106	1	0	0	3023	2935
02/23/22	3	14	3.5	3	817	813	1	0	0	3840	3748
02/24/22	2	13	3	2.75	979	971	2	-467	-467	4819	4719
02/25/22	3	11	2.75	3.5	0	0	0	0	0	4819	4719
02/28/22	3	11	2.75	3.5	136	132	1	0	0	4955	4851
03/01/22	2	13	3.25	3.5	0	0	0	0	0	4955	4851
03/02/22	3	13	3.5	3.5	414	410	1	0	0	5369	5261
03/03/22	3	18	2.25	3.5	0	0	0	0	0	5369	5261
03/04/22	3	18	2.25	3.5	0	0	0	0	0	5369	5261
03/07/22	3	18	2.25	3.5	0	0	0	0	0	5369	5261
03/08/22	3	18	2.25	3.5	(16)	(20)	1	-16	-16	5353	5241
03/09/22	3	18	2.25	3.5	414	410	1	0	0	5767	5651
03/10/22	1	6	3.5	3.5	(293)	(297)	1	-293	-293	5474	5354
03/11/22	2	13	2.75	1.75	532	528	1	0	0	6006	5882
03/14/22	2	13	2.75	1.75	479	475	1	0	0	6485	6357
03/15/22	3	18	1.75	3.25	789	785	1	0	0	7274	7142
03/16/22	1	18	1.75	3.5	621	617	1	0	0	7895	7759
03/17/22	1	13	1.75	3.5	0	0	0	0	0	7895	7759
03/18/22	1	15	1.5	2.5	476	472	1	0	0	8371	8231
03/21/22	1	9	2.5	3.5	0	0	0	0	0	8371	8231
03/22/22	1	9	2.5	3.5	414	410	1	0	0	8785	8641
03/23/22	3	11	3.25	3.5	51	47	1	0	0	8836	8688
03/24/22	1	9	0.75	3.5	500	496	1	0	0	9336	9184
03/25/22	2	12	0.5	3.25	(102)	(106)	1	-102	-102	9234	9078
03/28/22	1	7	2.25	3.5	385	381	1	0	0	9619	9459
03/29/22	1	11	2	3	133	129	1	0	0	9752	9588
03/30/22	1	10	2	3.25	0	0	0	0	0	9752	9588
03/31/22	3	17	2.5	3.5	0	0	0	0	0	9752	9588
04/01/22	3	17	2.5	3.5	0	0	0	0	0	9752	9588
04/04/22	3	17	2.5	3.5	595	591	1	0	0	10347	10179
04/05/22	3	16	2.75	3.5	0	0	0	0	0	10347	10179
04/06/22	2	10	2.75	3.5	(61)	(73)	3	-118	-206	10286	10106
04/07/22	2	20	3.5	2	0	0	0	0	0	10286	10106

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
04/08/22	2	10	3.25	3	228	224	1	0	0	10514	10330
04/11/22	1	5	3.25	1.5	335	331	1	0	0	10849	10661
04/12/22	2	12	3	2.5	(482)	(486)	1	-482	-482	10367	10175
04/13/22	3	15	3.5	1.5	0	0	0	0	0	10367	10175
04/14/22	3	16	3.5	1.5	563	559	1	0	0	10930	10734
04/18/22	3	18	3.5	1.5	204	200	1	0	0	11134	10934
04/19/22	2	14	3.25	2.75	0	0	0	0	0	11134	10934
04/20/22	2	15	3	1.75	200	196	1	0	0	11334	11130
04/21/22	1	5	1.25	2	325	317	2	-465	-465	11659	11447
04/22/22	2	6	3	2.5	649	645	1	0	0	12308	12092
04/25/22	2	6	3.25	2.5	(59)	(67)	2	-207	-207	12249	12025
04/26/22	2	8	3.5	3	765	761	1	0	0	13014	12786
04/27/22	1	6	3.25	2.75	(538)	(546)	2	-536	-538	12476	12240
04/28/22	1	8	3.25	2.75	0	0	0	0	0	12476	12240
04/29/22	1	8	3.25	2.75	1036	1032	1	0	0	13512	13272
05/02/22	1	11	3.25	2.25	97	93	1	0	0	13609	13365
05/03/22	1	12	3.5	2	0	0	0	0	0	13609	13365
05/04/22	1	12	3.5	2	429	425	1	0	0	14038	13790
05/05/22	1	18	1.25	2	919	915	1	0	0	14957	14705
05/06/22	1	13	1	2	(327)	(335)	2	-354	-354	14630	14370
05/09/22	1	12	1	2	(347)	(355)	2	-586	-586	14283	14015
05/10/22	1	7	1.75	3.5	(281)	(285)	1	-281	-281	14002	13730
05/11/22	1	9	1.75	3.25	0	0	0	0	0	14002	13730
05/12/22	2	17	3.5	1.5	(646)	(658)	3	-693	-833	13356	13072
05/13/22	1	17	3.5	0.5	399	391	2	0	0	13755	13463
05/16/22	1	20	3.25	1.5	0	0	0	0	0	13755	13463
05/17/22	1	20	3.25	1.5	0	0	0	0	0	13755	13463
05/18/22	1	13	3.5	1.75	1041	1037	1	0	0	14796	14500
05/19/22	1	8	3.25	3.25	0	0	0	0	0	14796	14500
05/20/22	1	9	3	3.25	112	108	1	0	0	14908	14608
05/23/22	1	18	3.25	1.75	215	211	1	0	0	15123	14819
05/24/22	1	8	3.5	3.5	(18)	(22)	1	-18	-18	15105	14797
05/25/22	1	6	3.5	3.25	0	0	0	0	0	15105	14797
05/26/22	1	6	3.5	3	0	0	0	0	0	15105	14797
05/27/22	2	16	3.25	2.75	476	472	1	0	0	15581	15269
05/31/22	3	12	3.25	3.25	0	0	0	0	0	15581	15269
06/01/22	3	16	2	3.5	(239)	(247)	2	-180	-239	15342	15022
06/02/22	3	19	3	3.5	0	0	0	0	0	15342	15022
06/03/22	3	19	3	3.5	161	157	1	0	0	15503	15179
06/06/22	2	9	3.5	2.25	(113)	(121)	2	-190	-190	15390	15058
06/07/22	1	9	3.25	0.75	(599)	(603)	1	-599	-599	14791	14455
06/08/22	2	13	3.5	1	42	38	1	0	0	14833	14493

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
06/09/22	3	19	3.25	1	515	511	1	0	0	15348	15004
06/10/22	2	15	3.5	0.5	(42)	(46)	1	-42	-42	15306	14958
06/13/22	3	20	3.25	1.25	145	141	1	0	0	15451	15099
06/14/22	1	7	3.5	3.5	0	0	0	0	0	15451	15099
06/15/22	2	16	2.5	3.5	310	306	1	0	0	15761	15405
06/16/22	1	20	2.25	2.25	441	437	1	0	0	16202	15842
06/17/22	2	18	2.25	2.25	(625)	(637)	3	-380	-697	15577	15205
06/21/22	2	17	3.25	2.25	42	38	1	0	0	15619	15243
06/22/22	1	5	2	3.25	11	7	1	0	0	15630	15250
06/23/22	1	7	1.25	3	0	0	0	0	0	15630	15250
06/24/22	1	7	1.5	2.5	508	504	1	0	0	16138	15754
06/27/22	1	7	1.5	2.5	0	0	0	0	0	16138	15754
06/28/22	2	9	3.5	2.75	887	883	1	0	0	17025	16637
06/29/22	3	16	2.75	3	(145)	(149)	1	-145	-145	16880	16488
06/30/22	3	15	2.25	2.5	345	337	2	0	0	17225	16825
07/01/22	2	12	2.75	2.75	(415)	(419)	1	-415	-415	16810	16406
07/05/22	2	16	2.75	2.25	37	29	2	-467	-467	16847	16435
07/06/22	2	16	2.75	3.5	0	0	0	0	0	16847	16435
07/07/22	2	16	1.5	3.5	235	231	1	0	0	17082	16666
07/08/22	2	6	3.5	3.25	0	0	0	0	0	17082	16666
07/11/22	2	6	3.5	3.25	345	341	1	0	0	17427	17007
07/12/22	2	7	3.25	3	(195)	(203)	2	-377	-377	17232	16804
07/13/22	2	7	3.25	2.5	31	23	2	-226	-226	17263	16827
07/14/22	3	10	3.25	3.5	(483)	(487)	1	-483	-483	16780	16340
07/15/22	3	15	3.25	3.5	0	0	0	0	0	16780	16340
07/18/22	3	15	3.25	3.5	0	0	0	0	0	16780	16340
07/19/22	3	15	3.25	3.5	576	572	1	0	0	17356	16912
07/20/22	3	14	3.5	3.5	0	0	0	0	0	17356	16912
07/21/22	3	14	3.5	3.5	0	0	0	0	0	17356	16912
07/22/22	1	12	0.25	3.5	(618)	(622)	1	-618	-618	16738	16290
07/25/22	1	9	0.5	2.75	(46)	(50)	1	-46	-46	16692	16240
07/26/22	3	15	2.75	3.5	0	0	0	0	0	16692	16240
07/27/22	3	15	2.75	1.75	(493)	(501)	2	-609	-609	16199	15739
07/28/22	3	18	2.5	2.5	0	0	0	0	0	16199	15739
07/29/22	3	10	3.5	3.5	0	0	0	0	0	16199	15739
08/01/22	3	7	2.75	3.25	(4)	(8)	1	-4	-4	16195	15731
08/02/22	2	5	2.75	3.5	0	0	0	0	0	16195	15731
08/03/22	3	9	0.75	3.5	614	610	1	0	0	16809	16341
08/04/22	3	11	0.5	3.5	(27)	(31)	1	-27	-27	16782	16310
08/05/22	3	11	2	3.25	(16)	(20)	1	-16	-16	16766	16290
08/08/22	3	11	2	3.25	0	0	0	0	0	16766	16290
08/09/22	3	11	2	3.25	0	0	0	0	0	16766	16290

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
08/10/22	3	8	2.5	2.5	340	328	3	0	0	17106	16618
08/11/22	1	5	3.5	2.75	0	0	0	0	0	17106	16618
08/12/22	1	8	3.25	2.5	0	0	0	0	0	17106	16618
08/15/22	1	7	3.25	2.5	0	0	0	0	0	17106	16618
08/16/22	1	9	3	2.5	0	0	0	0	0	17106	16618
08/17/22	1	9	3	2.5	31	27	1	0	0	17137	16645
08/18/22	1	5	2.25	3	0	0	0	0	0	17137	16645
08/19/22	1	5	2.25	3	274	270	1	0	0	17411	16915
08/22/22	1	12	2	2.5	398	394	1	0	0	17809	17309
08/23/22	2	12	1.75	2.75	0	0	0	0	0	17809	17309
08/24/22	2	7	3	2.25	0	0	0	0	0	17809	17309
08/25/22	2	8	2.75	1.75	0	0	0	0	0	17809	17309
08/26/22	2	8	2.75	1.75	919	915	1	0	0	18728	18224
08/29/22	1	5	2	1.5	117	113	1	0	0	18845	18337
08/30/22	3	13	2.75	1	253	249	1	0	0	19098	18586
08/31/22	2	9	2.5	0.5	153	145	2	-120	-120	19251	18731
09/01/22	3	8	3.25	0.75	(185)	(193)	2	-161	-185	19066	18538
09/02/22	3	8	3.5	0.75	477	473	1	0	0	19543	19011
09/06/22	3	8	3.5	1	173	169	1	0	0	19716	19180
09/07/22	3	6	3	3.5	494	490	1	0	0	20210	19670
09/08/22	3	16	1.75	3.5	135	131	1	0	0	20345	19801
09/09/22	3	16	1.75	3.5	333	329	1	0	0	20678	20130
09/12/22	3	20	2.25	3.25	0	0	0	0	0	20678	20130
09/13/22	3	20	2.25	3.25	(534)	(542)	2	-680	-680	20144	19588
09/14/22	1	7	3.25	2.75	0	0	0	0	0	20144	19588
09/15/22	1	7	2.75	2.75	0	0	0	0	0	20144	19588
09/16/22	1	7	2	2.5	(274)	(278)	1	-274	-274	19870	19310
09/19/22	2	10	3.5	2.5	0	0	0	0	0	19870	19310
09/20/22	2	10	3.5	2.5	0	0	0	0	0	19870	19310
09/21/22	1	5	2.75	3.5	(249)	(261)	3	-250	-490	19621	19049
09/22/22	1	14	3.5	2.5	0	0	0	0	0	19621	19049
09/23/22	2	6	1.75	2	158	154	1	0	0	19779	19203
09/26/22	2	7	1.75	3	(35)	(39)	1	-35	-35	19744	19164
09/27/22	2	7	1.75	3	522	510	3	-96	-96	20266	19674
09/28/22	1	11	2.75	3.5	0	0	0	0	0	20266	19674
09/29/22	2	7	2.25	3.25	182	174	2	-92	-92	20448	19848
09/30/22	2	14	3.5	2.25	0	0	0	0	0	20448	19848
10/03/22	2	9	2	3.5	0	0	0	0	0	20448	19848
10/04/22	1	9	3.5	1	(125)	(133)	2	-151	-151	20323	19715
10/05/22	1	10	1.5	1.5	(376)	(380)	1	-376	-376	19947	19335
10/06/22	1	6	2	3.25	0	0	0	0	0	19947	19335
10/07/22	3	13	3	3.25	586	582	1	0	0	20533	19917

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
10/10/22	3	12	3	3.25	0	0	0	0	0	20533	19917
10/11/22	3	12	3	3.25	0	0	0	0	0	20533	19917
10/12/22	2	6	3	1.25	57	53	1	0	0	20590	19970
10/13/22	1	19	2.75	3.5	143	139	1	0	0	20733	20109
10/14/22	2	11	2.75	2.5	542	538	1	0	0	21275	20647
10/17/22	1	6	1.75	2	204	200	1	0	0	21479	20847
10/18/22	1	5	2.25	3.5	(474)	(478)	1	-474	-474	21005	20369
10/19/22	1	6	1.75	2.75	(308)	(312)	1	-308	-308	20697	20057
10/20/22	1	6	1.75	2.75	(335)	(339)	1	-335	-335	20362	19718
10/21/22	1	6	2.25	2.75	0	0	0	0	0	20362	19718
10/24/22	1	6	2.25	2.75	0	0	0	0	0	20362	19718
10/25/22	1	13	3.5	2.25	0	0	0	0	0	20362	19718
10/26/22	1	13	3.5	2	3	(1)	1	0	0	20365	19717
10/27/22	1	16	3	1.75	206	202	1	0	0	20571	19919
10/28/22	1	5	0.5	3.5	658	654	1	0	0	21229	20573
10/31/22	1	11	2	2.5	0	0	0	0	0	21229	20573
11/01/22	1	11	2	2.5	177	173	1	0	0	21406	20746
11/02/22	3	8	2.25	2.5	(5)	(25)	5	-302	-302	21401	20721
11/03/22	2	11	3.5	1.75	206	202	1	0	0	21607	20923
11/04/22	1	6	3.25	1	70	66	1	0	0	21677	20989
11/07/22	2	9	3.25	2	0	0	0	0	0	21677	20989
11/08/22	2	7	3.5	2.5	(24)	(28)	1	-24	-24	21653	20961
11/09/22	2	9	3.25	2.5	355	351	1	0	0	22008	21312
11/10/22	2	7	3.5	2.5	(450)	(458)	2	-580	-580	21558	20854
11/11/22	1	5	2.75	2.75	0	0	0	0	0	21558	20854
11/14/22	1	5	2	3.25	0	0	0	0	0	21558	20854
11/15/22	1	5	2	3.25	(330)	(334)	1	-330	-330	21228	20520
11/16/22	1	14	2.5	3.25	0	0	0	0	0	21228	20520
11/17/22	1	16	1	2.5	(72)	(76)	1	-72	-72	21156	20444
11/18/22	1	15	1.75	2.75	0	0	0	0	0	21156	20444
11/21/22	1	15	1.75	2.75	0	0	0	0	0	21156	20444
11/22/22	1	15	1.75	2.75	0	0	0	0	0	21156	20444
11/23/22	3	18	0.75	2.75	128	124	1	0	0	21284	20568
11/25/22	3	13	1.5	2.75	0	0	0	0	0	21284	20568
11/28/22	2	11	1.25	3.5	0	0	0	0	0	21284	20568
11/29/22	1	6	1	3.25	0	0	0	0	0	21284	20568
11/30/22	1	6	1	2.25	1068	1064	1	0	0	22352	21632
12/01/22	1	7	2	1.75	(540)	(548)	2	-300	-540	21812	21084
12/02/22	2	6	1.5	3.25	442	434	2	0	0	22254	21518
12/05/22	3	16	0.25	3.5	(257)	(261)	1	-257	-257	21997	21257
12/06/22	3	16	0.25	3.5	(208)	(212)	1	-208	-208	21789	21045
12/07/22	3	9	3.5	1.75	84	80	1	0	0	21873	21125

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
12/08/22	3	7	3.5	1.25	5	1	1	0	0	21878	21126
12/09/22	3	8	3.5	2	212	208	1	0	0	22090	21334
12/12/22	2	9	3	0.5	(291)	(295)	1	-291	-291	21799	21039
12/13/22	2	7	3.25	0.75	872	864	2	0	0	22671	21903
12/14/22	2	11	3.25	3.5	(101)	(105)	1	-101	-101	22570	21798
12/15/22	2	14	3	3.5	0	0	0	0	0	22570	21798
12/16/22	1	16	2	3	0	0	0	0	0	22570	21798
12/19/22	1	6	2.75	3.25	0	0	0	0	0	22570	21798
12/20/22	1	6	2.75	3.25	0	0	0	0	0	22570	21798
12/21/22	1	6	2.75	3.25	0	0	0	0	0	22570	21798
12/22/22	1	6	2.75	3.25	153	149	1	0	0	22723	21947
12/23/22	1	6	2.75	3.25	0	0	0	0	0	22723	21947
12/27/22	1	6	2.75	3.25	0	0	0	0	0	22723	21947
12/28/22	2	19	3	2.75	0	0	0	0	0	22723	21947
12/29/22	3	18	0.75	3.5	382	378	1	0	0	23105	22325
12/30/22	2	19	1.75	3.5	97	93	1	0	0	23202	22418
01/03/23	1	6	2.25	1.25	(197)	(205)	2	-292	-292	23005	22213
01/04/23	3	8	3.25	3	0	0	0	0	0	23005	22213
01/05/23	3	8	3.25	3	161	157	1	0	0	23166	22370
01/06/23	2	6	2.5	2.75	374	362	3	-308	-351	23540	22732
01/09/23	2	8	2.75	3.25	0	0	0	0	0	23540	22732
01/10/23	2	8	2.75	3.25	0	0	0	0	0	23540	22732
01/11/23	3	11	3.5	2.25	0	0	0	0	0	23540	22732
01/12/23	3	11	3.5	2.25	(382)	(386)	1	-382	-382	23158	22346
01/13/23	1	6	1.5	3.5	0	0	0	0	0	23158	22346
01/17/23	1	6	1.5	3.5	(140)	(144)	1	-140	-140	23018	22202
01/18/23	3	6	3.25	3.25	0	0	0	0	0	23018	22202
01/19/23	2	5	3.5	3.5	0	0	0	0	0	23018	22202
01/20/23	2	5	3.5	3.5	0	0	0	0	0	23018	22202
01/23/23	2	6	3.5	3	0	0	0	0	0	23018	22202
01/24/23	3	20	1.5	2.5	0	0	0	0	0	23018	22202
01/25/23	3	20	1.5	2.5	719	711	2	0	0	23737	22913
01/26/23	2	19	1.25	3.5	71	67	1	0	0	23808	22980
01/27/23	3	15	0.75	3.5	228	224	1	0	0	24036	23204
01/30/23	3	15	1.25	3.5	(341)	(345)	1	-341	-341	23695	22859
01/31/23	3	15	1.25	3.5	0	0	0	0	0	23695	22859
02/01/23	1	13	1.5	2.5	170	166	1	0	0	23865	23025
02/02/23	1	18	3.25	2.25	214	210	1	0	0	24079	23235
02/03/23	1	20	3	2.75	0	0	0	0	0	24079	23235
02/06/23	1	20	3	2.75	0	0	0	0	0	24079	23235
02/07/23	1	20	3	2.75	0	0	0	0	0	24079	23235
02/08/23	1	20	3	2.75	0	0	0	0	0	24079	23235

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
02/09/23	2	17	3.5	1	409	405	1	0	0	24488	23640
02/10/23	2	20	3.5	0.75	(107)	(111)	1	-107	-107	24381	23529
02/13/23	2	20	3.5	0.75	(44)	(48)	1	-44	-44	24337	23481
02/14/23	2	20	3.5	1.25	(395)	(399)	1	-395	-395	23942	23082
02/15/23	3	16	3.5	0.75	(248)	(252)	1	-248	-248	23694	22830
02/16/23	2	13	3	1.25	112	108	1	0	0	23806	22938
02/17/23	3	9	3.5	2.25	86	82	1	0	0	23892	23020
02/21/23	1	11	2.75	1.5	365	361	1	0	0	24257	23381
02/22/23	3	10	2.75	2.75	0	0	0	0	0	24257	23381
02/23/23	3	10	2.75	2.75	(50)	(54)	1	-50	-50	24207	23327
02/24/23	1	6	3.5	2	102	98	1	0	0	24309	23425
02/27/23	1	7	2	2	(117)	(121)	1	-117	-117	24192	23304
02/28/23	3	15	3.25	0.25	1	(3)	1	0	0	24193	23301
03/01/23	1	9	3	2.5	0	0	0	0	0	24193	23301
03/02/23	1	9	3	2.5	0	0	0	0	0	24193	23301
03/03/23	2	15	3	2	0	0	0	0	0	24193	23301
03/06/23	1	5	1.75	2.75	(127)	(131)	1	-127	-127	24066	23170
03/07/23	1	5	2	3.25	95	91	1	0	0	24161	23261
03/08/23	1	5	2	3.5	0	0	0	0	0	24161	23261
03/09/23	1	5	2	3.5	0	0	0	0	0	24161	23261
03/10/23	1	8	1.75	3.5	0	0	0	0	0	24161	23261
03/13/23	3	6	3.25	1.75	(233)	(245)	3	-175	-233	23928	23016
03/14/23	3	16	3	1.25	(39)	(47)	2	-32	-39	23889	22969
03/15/23	2	20	2.75	0.25	(162)	(166)	1	-162	-162	23727	22803
03/16/23	2	13	2	1.5	440	436	1	0	0	24167	23239
03/17/23	2	20	1.5	3.25	0	0	0	0	0	24167	23239
03/20/23	2	20	1.5	3.25	0	0	0	0	0	24167	23239
03/21/23	1	12	1.75	2.25	226	222	1	0	0	24393	23461
03/22/23	2	14	1.25	2.75	0	0	0	0	0	24393	23461
03/23/23	2	13	1.25	3	12	8	1	0	0	24405	23469
03/24/23	1	10	1.25	2.75	109	105	1	0	0	24514	23574
03/27/23	1	13	1.75	2.75	0	0	0	0	0	24514	23574
03/28/23	1	13	1.25	2	0	0	0	0	0	24514	23574
03/29/23	3	16	3.5	0.5	(116)	(124)	2	-136	-136	24398	23450
03/30/23	1	15	2.5	1.75	0	0	0	0	0	24398	23450
03/31/23	3	15	2	3.25	0	0	0	0	0	24398	23450
04/03/23	3	15	2	3.25	0	0	0	0	0	24398	23450
04/04/23	3	15	2	3.25	0	0	0	0	0	24398	23450
04/05/23	1	13	1	3.25	0	0	0	0	0	24398	23450
04/06/23	1	18	1.75	1.5	0	0	0	0	0	24398	23450
04/10/23	2	8	1	3.25	414	406	2	0	0	24812	23856
04/11/23	2	15	1.25	2	0	0	0	0	0	24812	23856

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
04/12/23	2	16	1	2	(246)	(250)	1	-246	-246	24566	23606
04/13/23	3	7	3.25	2.75	351	347	1	0	0	24917	23953
04/14/23	3	7	2.75	2.75	0	0	0	0	0	24917	23953
04/17/23	3	7	2.75	2.75	0	0	0	0	0	24917	23953
04/18/23	3	7	2.75	2.75	0	0	0	0	0	24917	23953
04/19/23	3	7	2.75	2.75	0	0	0	0	0	24917	23953
04/20/23	3	7	2.75	2.75	61	57	1	0	0	24978	24010
04/21/23	3	8	2.5	3.25	0	0	0	0	0	24978	24010
04/24/23	3	8	2.5	3.25	0	0	0	0	0	24978	24010
04/25/23	3	8	2.5	2.25	0	0	0	0	0	24978	24010
04/26/23	3	8	2.5	2.25	(77)	(81)	1	-77	-77	24901	23929
04/27/23	3	9	3	0.25	(492)	(500)	2	-508	-508	24409	23429
04/28/23	2	6	2.25	2.75	0	0	0	0	0	24409	23429
05/01/23	2	6	2.25	2.75	0	0	0	0	0	24409	23429
05/02/23	2	6	2.25	2.75	0	0	0	0	0	24409	23429
05/03/23	3	7	2	3.5	0	0	0	0	0	24409	23429
05/04/23	3	7	2	3.5	0	0	0	0	0	24409	23429
05/05/23	1	12	1.5	1.75	364	360	1	0	0	24773	23789
05/08/23	1	20	0.75	1.75	0	0	0	0	0	24773	23789
05/09/23	1	16	1	1.75	0	0	0	0	0	24773	23789
05/10/23	1	16	1	1.75	164	160	1	0	0	24937	23949
05/11/23	2	8	1.5	1.75	0	0	0	0	0	24937	23949
05/12/23	3	19	1.5	1.5	0	0	0	0	0	24937	23949
05/15/23	2	20	0.75	2	90	86	1	0	0	25027	24035
05/16/23	1	6	1	2.25	0	0	0	0	0	25027	24035
05/17/23	1	6	1	2.25	0	0	0	0	0	25027	24035
05/18/23	2	14	0.75	1.25	429	425	1	0	0	25456	24460
05/19/23	3	17	1	1.25	0	0	0	0	0	25456	24460
05/22/23	3	12	1.25	2.75	0	0	0	0	0	25456	24460
05/23/23	3	11	1.25	2.25	0	0	0	0	0	25456	24460
05/24/23	2	5	2	3.25	0	0	0	0	0	25456	24460
05/25/23	3	10	1.75	1	185	173	3	-71	-88	25641	24633
05/26/23	3	7	2	3.25	755	751	1	0	0	26396	25384
05/30/23	2	7	1.5	3.5	(406)	(418)	3	-192	-406	25990	24966
05/31/23	2	5	2.25	2	0	0	0	0	0	25990	24966
06/01/23	2	5	2.25	2	0	0	0	0	0	25990	24966
06/02/23	2	5	1.75	2	(38)	(50)	3	-188	-188	25952	24916
06/05/23	1	16	1.75	1.25	0	0	0	0	0	25952	24916
06/06/23	1	20	1.25	1.25	0	0	0	0	0	25952	24916
06/07/23	2	6	2	2.25	621	613	2	-106	-106	26573	25529
06/08/23	3	18	3.5	1.5	0	0	0	0	0	26573	25529
06/09/23	3	17	3.5	1.5	0	0	0	0	0	26573	25529

OSS Date	IS pw	IS N	IS vup	IS vdn	osgp	osnp\$4	ont	ollt	odd	OSS EQ	OSS NetEq
06/12/23	3	17	3.5	1.5	0	0	0	0	0	26573	25529
06/13/23	2	16	1	2	(46)	(50)	1	-46	-46	26527	25479
06/14/23	3	19	1.25	2.75	(1)	(5)	1	-1	-1	26526	25474
06/15/23	2	15	1	2.25	420	416	1	0	0	26946	25890
06/16/23	1	5	1.25	2.75	0	0	0	0	0	26946	25890
06/20/23	2	6	2	3.5	0	0	0	0	0	26946	25890
06/21/23	2	6	1.25	3.5	55	51	1	0	0	27001	25941
06/22/23	2	16	1.25	2	189	185	1	0	0	27190	26126
06/23/23	2	16	1.25	2.75	(20)	(28)	2	-67	-67	27170	26098
06/26/23	2	7	3.5	1.75	118	114	1	0	0	27288	26212
06/27/23	2	5	2	3.25	422	418	1	0	0	27710	26630
06/28/23	2	5	2.5	3	0	0	0	0	0	27710	26630
06/29/23	3	6	2.5	2.5	0	0	0	0	0	27710	26630
06/30/23	2	14	2.5	1.5	234	230	1	0	0	27944	26860
07/03/23	2	11	2.5	2.5	0	0	0	0	0	27944	26860
07/05/23	2	11	2.5	2.5	0	0	0	0	0	27944	26860
07/06/23	2	11	2.5	3.5	(107)	(111)	1	-107	-107	27837	26749
07/07/23	3	20	1.5	3.25	0	0	0	0	0	27837	26749
07/10/23	3	7	1.75	3	0	0	0	0	0	27837	26749
07/11/23	1	9	1.5	2	0	0	0	0	0	27837	26749
07/12/23	1	9	1.25	2	57	53	1	0	0	27894	26802
07/13/23	3	7	0.25	3.25	322	318	1	0	0	28216	27120
07/14/23	1	13	1.75	3.25	0	0	0	0	0	28216	27120
07/17/23	2	10	1.5	3.5	221	217	1	0	0	28437	27337
07/18/23	2	10	1.25	3.5	288	284	1	0	0	28725	27621
07/19/23	2	10	1.75	3.5	0	0	0	0	0	28725	27621
07/20/23	2	10	1.75	3.5	0	0	0	0	0	28725	27621
07/21/23	2	19	3.25	1.5	30	26	1	0	0	28755	27647
07/24/23	2	14	3.5	1.5	(77)	(81)	1	-77	-77	28678	27566
07/25/23	2	16	3.5	1.75	0	0	0	0	0	28678	27566
07/26/23	2	16	3.5	1.75	36	32	1	0	0	28714	27598
07/27/23	1	10	3	1.25	272	264	2	-247	-247	28986	27862
07/28/23	2	14	1.75	2.25	227	223	1	0	0	29213	28085
07/31/23	2	17	1.75	3.5	0	0	0	0	0	29213	28085
08/01/23	2	17	1.75	3.5	0	0	0	0	0	29213	28085
08/02/23	2	16	1.25	2	(60)	(68)	2	-209	-209	29153	28017
08/03/23	1	5	2.25	3	0	0	0	0	0	29153	28017
08/04/23	2	11	1.5	3	(559)	(563)	1	-559	-559	28594	27454

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 5-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:

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

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

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 this type of matrix equation solvers is very slow and for these types of problems are unstable. They cause numerical errors and floating-point overflows due to matrix inversion ill conditioning which produces false results.

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

Fortunately, these types of problems can be solved by a fast, efficient and accurate algorithm using Discrete Orthogonal Legendre Polynomials. This method is explained in detail in Norman Morrison's 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

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

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

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

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** = $d^2P_F(T+1)/d^2t$ 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 5-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.

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

$$\text{Acceleration}(t+1) = d^2P_f / d^2t = 2 c_0 + 9d_0 * (T+1) + 12e_0 * (T+1)^3 + \dots + n * (n-1) * b_n * (T+1)^{n-2}$$

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 (SD) 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 SD is 6.8 times the SD for degree=1, $N=20$. This creates problems when trying to determine the correct ranges for vup/vdn and aup/adn during optimization.

@ES.D 5 min bars Date Range 1140801 to 1150220
Total Number of Bars=56340 Norm=0
FixmVn Multiplier= 1/SD to Scale Velocity pw and N Range to One SD

Degree	N	SD	1/SD
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	9.3890
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	0.1583	3.8275
2	60	0.2371	4.2180
2	70	0.2173	4.6010

The Normalization Multiplier

avg		0.2981	3.5425
3	20	0.7854	1.2732
3	30	0.5933	1.6855
3	40	0.4973	2.0111
3	50	0.4347	2.3005
3	60	0.3949	2.5324
3	70	0.3656	2.7352
avg		0.5119	2.0897
4	20	1.2924	0.7738
4	30	0.9279	1.0777
4	40	0.7582	1.3189
4	5	0.6542	1.5285
4	60	0.5804	1.7228
4	70	0.5314	1.8818
avg		0.7908	1.3839

The problem may get worse when we want to find good inputs for other tradables. Other tradables, because of their scales and tick size have much different Velocity ranges 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 and Acceleration 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.