Gaussian and Other Low Lag Filters
John Ehlers

The first objective of using smoothers is to eliminate or reduce the undesired high frequency components in the price data. Therefore, these smoothers are called low pass filters, and they all work by averaging in one way or another. I previously described[1] the design and use of Butterworth low pass filters to achieve greater filtering than can be obtained by simple averagers. However, nothing comes for free. A higher degree of filtering is necessarily accompanied by a larger amount of lag. That's just a fact of life.

Since lag is the downfall of most trading indicators, leading to the failure to react to price changes in a timely manner, a better approach to filtering is to minimize the lag and accept whatever smoothing results. The importance of lag is demonstrated in Figure 1, the lag of a 3 pole Butterworth filter that attenuates cycles shorter than 10 bars. We traders think in terms of cycle periods, but filter responses are usually plotted in terms of frequency. Frequency is the reciprocal of period. The frequency scale is normalized to a 2 bar cycle (the Nyquist frequency for daily data).
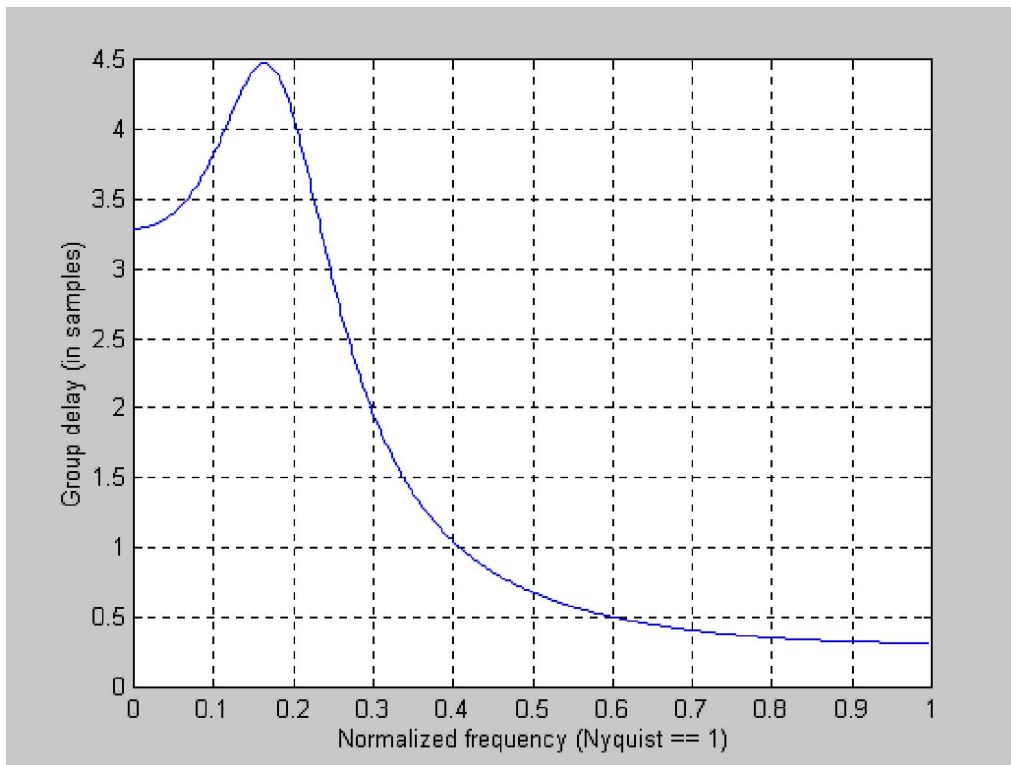

Figure 1. Lag of a 3 Pole Butterworth Filter with a 10 bar Period Cutoff

---

[1] John Ehlers "Poles, Zeros, and Higher Order Filters"

The low frequency lag of a Butterworth filter can be estimated from the equation

$$Lag = N*P / \pi^2$$

Where N is the number of poles in the filter and P is the longest cycle period to be passed through the filter. The lag story gets worse as the frequency components of the input waveform are near the band edge of the filter. The higher frequency components within the passband of the filter are actually delayed *more* than the lower frequency components. This is exactly the opposite of what a trader desires. We have to react faster to faster changes in the market, and so we would prefer a smoothing filter that actually has less lag to the higher frequency components.

The use of Gaussian filters is a move toward the dual goals of reducing lag and reducing the lag of high frequency components relative to the lag of lower frequency components. Multipole Gaussian filters can be constructed that provide a desired degree of smoothing. The group delay of a 3 pole Gaussian filter having a .1 cycle per day passband is shown in Figure 2 for comparison to the delay produced by a Butterworth filter.
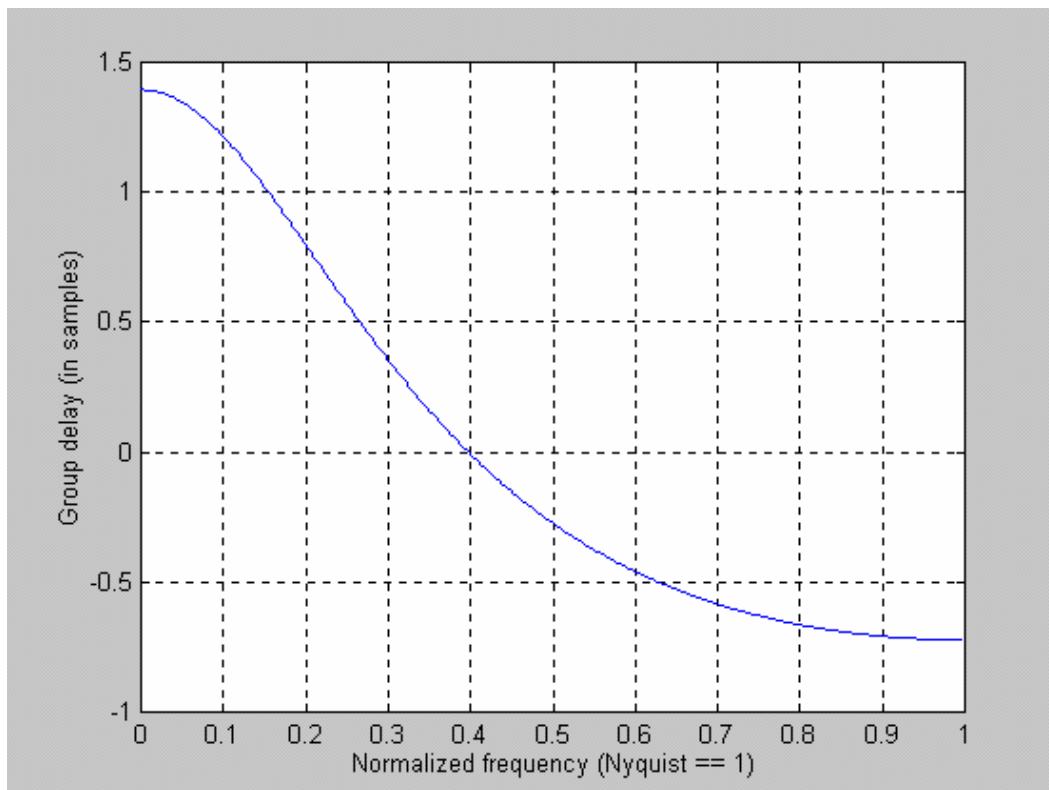

Figure 2. Lag of a 3 Pole Gaussian Filter with a 10 Bar period Cutoff

For an equivalent number of poles, the lag of a Gaussian filter is about half the lag of a Butterworth filter. More importantly, the higher frequency components have still less lag than the low frequency components. With Gaussian filters the lag as a function of frequency is going in the right direction for traders. It is also true that a Gaussian filter has about half the smoothing effectiveness as an equivalently sized Butterworth filters.

Said another way, a 4 pole Gaussian filter has about the same smoothing performance as a 2 pole Butterworth filter. Thus, to do the same filtering job, these two filters would have about the same low frequency lag but the Gaussian filter would preserve the original price function with greater fidelity because the higher frequency components within the passband would not be delayed as much as in the Butterworth filter. Comparative filter responses of a 2 pole Butterworth filter and a 2 pole Gaussian filter, each having a 10 bar cycle passband, is shown in Figure 3.
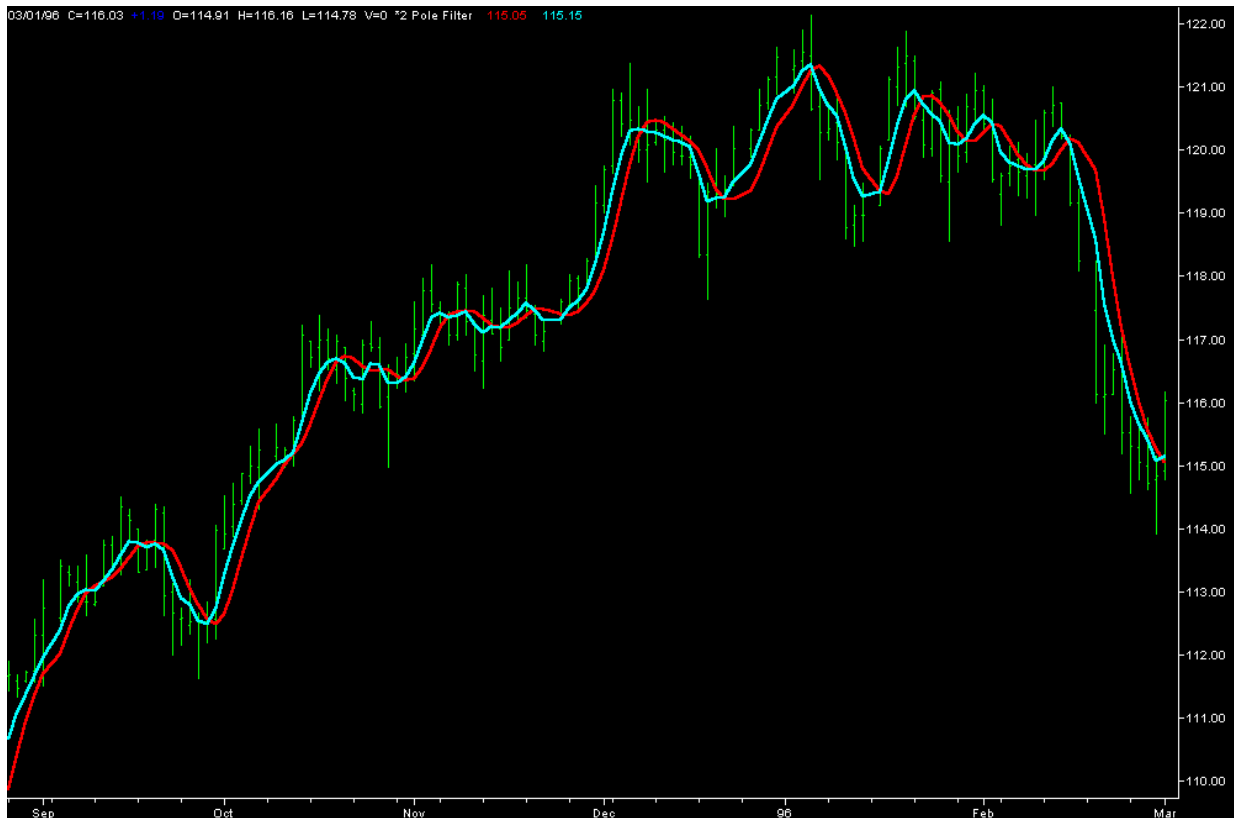

Figure 3. Comparison of 2 Pole Filters shows the Gaussian filter (Cyan)has much less lag than the Butterworth filter (Red). The Gaussian filter has less smoothing.

NUTS AND BOLTS

There is no magic to a Gaussian filter. It is just the multiple application of an Exponential Moving Average. From my previous article[2] the transfer response of an exponential moving average is

$$H(z) = \alpha / (1 - (1-\alpha) Z^{-1})$$

So that applying the exponential moving average "N" times gives a N Pole filter response as

$$H(z) = \alpha^N / (1 - (1-\alpha) Z^{-1})^N$$

---

[2] ibid

At zero frequency $Z^{-1}=1$, so this low pass filter has unity gain. Also, the denominator assumes the value of α^N at zero frequency. The corner frequency of the filter is defined as that point where the transfer response is down by 3 dB, or .707 in amplitude. When this occurs, we have the relationship

$$(1 - (1-\alpha) Z^{-1})^N = 1.414\alpha^N \quad \text{where } Z^{-1} = e^{-j\omega} \text{ and } \omega = 2\pi/P$$

Crunching through the complex arithmetic, we arrive at the solution for alpha as

$$\alpha = -\beta + \text{SQR}(\beta^2 + 2\beta)$$

$$\text{Where } \beta = (1 - \cos(\omega)) / (1.414^{2/N} - 1)$$

This generalized solution for alpha can be used to compute the coefficients for any order Gaussian filter. Recalling that the filtered output is determined by the equation

$$f(z) = H(z)g(z) \quad \text{where } g(z) \text{ is the input price}$$

If $H(z)$ is of the form $1/(1 - (1-\alpha) Z^{-1})^N$ we can easily form equations for the output in EasyLanguage metaphor because Z^{-1} is synonymous with a one bar lag. These equations are:

One Pole: $f = \alpha g + (1-\alpha)f[1]$

Two Poles: $f = \alpha^2 g + 2(1-\alpha)f[1] - (1-\alpha)^2 f[2]$

Three Poles: $f = \alpha^3 g + 3(1-\alpha)f[1] - 3(1-\alpha)^2 f[2] + (1-\alpha)^3 f[3]$

Four Poles: $f = \alpha^4 g + 4(1-\alpha)f[1] - 6(1-\alpha)^2 f[2] + 4(1-\alpha)^3 f[3] - (1-\alpha)^4 f[4]$

Etc.

GAUSSIAN FILTER TABLES

It is often easier to go to a lookup table to get filter coefficients rather than uniquely calculate the coefficients each time they are used. In the following tables, the notation is that B is used with the price data and A is used with the previously calculated filter output [N] bars ago. I hope these tables at the end of the article make it easier to use higher order filters.

REMOVING LAG TECHNIQUES

The equation for a simple 3 bar moving average is

$$f = .25*g + .5*g[1] + .25*g[2]$$

where each of the g's corresponds to the price. In terms of navigation, the g values are the values of position to compute a smoothed estimate of position. If we take a page from the book on Kalman filters and introduce a velocity term in addition to the position term we can arrive at a better smoothed estimate. So, in the above equation, let each of the price values become $g + (g - g[1]) = 2*g - g[1]$. The three bar moving average equation then becomes

$$f = .5*g + .75g[1] + 0 - .25*g[3]$$

Guaranteed you will not get much filtering out of this filter. In fact, you will probably get some overshoot. Its strength is that you can filter one of the Gaussian filter smoothed results to further reduce the higher frequency lag. Remember, that by decreasing the lag you are also decreasing the amount of smoothing that you can obtain.

One Pole (EMA)

Period	B[0]	A[1]
2	0.828427	0.171573
4	0.732051	0.267949
6	0.618034	0.381966
8	0.526602	0.473398
10	0.455887	0.544113
12	0.400720	0.599280
14	0.356896	0.643104
16	0.321416	0.678584
18	0.292186	0.707814
20	0.267730	0.732270
22	0.246990	0.753010
24	0.229192	0.770808
26	0.213760	0.786240
28	0.200256	0.799744
30	0.188343	0.811657
32	0.177759	0.822241
34	0.168294	0.831706
36	0.159780	0.840220
38	0.152082	0.847918
40	0.145089	0.854911

Two Pole

Period	B[0]	A[1]	A[2]
2	0.834615	0.172854	-0.007470
4	0.722959	0.299460	-0.022419
6	0.578300	0.479080	-0.057379
8	0.457577	0.647112	-0.104688
10	0.365017	0.791668	-0.156684
12	0.295336	0.913103	-0.208439
14	0.242632	1.014847	-0.257479
16	0.202250	1.100556	-0.302806
18	0.170835	1.173357	-0.344192
20	0.146017	1.235757	-0.381774
22	0.126125	1.289719	-0.415844
24	0.109966	1.336777	-0.446743
26	0.096680	1.378133	-0.474813
28	0.085633	1.414738	-0.500371
30	0.076357	1.447346	-0.523703
32	0.068496	1.476567	-0.545063
34	0.061779	1.502894	-0.564672
36	0.055996	1.526729	-0.582726
38	0.050984	1.548408	-0.599392
40	0.046612	1.568205	-0.614817

Three Pole

Period	B[0]	A[1]	A[2]	A[3]
2	0.836701	0.173094	-0.009987	0.000192
4	0.718670	0.312814	-0.032617	0.001134
6	0.558792	0.529009	-0.093283	0.005483
8	0.422292	0.749259	-0.187130	0.015579
10	0.318295	0.951680	-0.301899	0.031923
12	0.242068	1.130321	-0.425875	0.053486
14	0.186612	1.285644	-0.550960	0.078704
16	0.146016	1.420251	-0.672371	0.106104
18	0.115940	1.537154	-0.787614	0.134520
20	0.093340	1.639147	-0.895601	0.163114
22	0.076111	1.728632	-0.996056	0.191313
24	0.062791	1.807607	-1.089148	0.218750
26	0.052354	1.877714	-1.175270	0.245202
28	0.044075	1.940297	-1.254918	0.270546
30	0.037432	1.996460	-1.328618	0.294726
32	0.032045	2.047111	-1.396887	0.317731
34	0.027635	2.093000	-1.460217	0.339582
36	0.023991	2.134754	-1.519058	0.360313
38	0.020956	2.172895	-1.573824	0.379973
40	0.018409	2.207865	-1.624889	0.398615

Four
Pole

Period	B[0]	A[1]	A[2]	A[3]	A[4]
2	0.837747	0.173178	-0.011247	0.000325	0.000004
4	0.716200	0.320247	-0.038459	0.002053	0.000041
6	0.547128	0.559812	-0.117521	0.010965	0.000384
8	0.400596	0.817734	-0.250758	0.034176	0.001747
10	0.289459	1.066023	-0.426152	0.075715	0.005045
12	0.209659	1.293310	-0.627244	0.135204	0.010929
14	0.153408	1.496649	-0.839984	0.209527	0.019599
16	0.113779	1.676861	-1.054449	0.294694	0.030885
18	0.085632	1.836187	-1.264344	0.386929	0.044405
20	0.065397	1.977213	-1.466015	0.483104	0.059700
22	0.050648	2.102418	-1.657560	0.580814	0.076320
24	0.039744	2.214012	-1.838193	0.678297	0.093860
26	0.031571	2.313903	-2.007804	0.774311	0.111980
28	0.025363	2.403709	-2.166681	0.868012	0.130403
30	0.020589	2.484797	-2.315331	0.958854	0.148910
32	0.016875	2.558316	-2.454368	1.046508	0.167331
34	0.013953	2.625237	-2.584450	1.130799	0.185538
36	0.011632	2.686378	-2.706235	1.211662	0.203436
38	0.009770	2.742435	-2.820356	1.289107	0.220956
40	0.008263	2.794000	-2.927413	1.363199	0.238049