Ehlers Filters

by
John Ehlers

The most common filters used by traders are Moving Averages – either Simple Moving Averages (SMA) or Exponential Moving Averages (EMA). These are linear filters. Linear filters are optimal for smoothing stationary, slowly varying signals that are corrupted with high frequency noise. Unfortunately, price data is not stationary much of the time. A coin flip experiment is an example of a statistical stationary process. However, if weighted coins are introduced into the experiment randomly, the statistics of the experiment now depend on which coin is used, and therefore are nonstationary. The signals we deal with can often be described statistically. For example, human speech has noise-like statistics. The process is nonstationary because it changes from moment to moment. Although speech has noise-like characteristics, that is not to say that it does not carry information (unless coming from a lawyer or politician). Price data resembles speech in statistical characteristics. It is both noise-like and nonstationary. One of the main problems we encounter in trading when using technical analysis is that we must attempt to restore signals that are often nonstationary and are also corrupted by noise. When dealing with nonstationary signals that have sharp transitions of their mean or when dealing with impulsive noise, linear filtering techniques give poor results. In this article I will describe how you can make some amazing nonlinear filters that better handles these signals.

There are two basic kinds of filters, Finite Impulse Response (FIR) Filters and Infinite Impulse Response (IIR) Filters. An EMA is an example of an IIR filter. An impulse the mathematical equivalent of a sharp spike. In sampled data, the impulse has a value of one for only one sample and the data is zero for all other samples. In contrast, when an impulse is applied to an IIR Filter, a portion of the first output is fed back to the input and added to the next data input sample. Because this calculation is iterative, the effects of the impulse is theoretically present in the output forever – hence the name IIR. When the impulse is applied to the input of an FIR Filter, a SMA for example, the effects of the impulse will be present in the filter output only over the length of the filter and the output will be zero otherwise – hence the name FIR.

Perry Kaufman, Tushar Chande (see the sidebar) and others have designed nonlinear IIR filters to better smooth market data. The basic approach is to craft a volatility-adjusted value for the alpha parameter in an EMA. These are certainly workable approaches, but their effectiveness is constrained by the opposing requirements of providing adequate smoothing while vigorously attacking major price movements.

The filters I have invented are nonlinear FIR filters. It turns out that they provide both extraordinary smoothing in sideways markets and aggressively follow major price movements with minimal lag. The development of my filters starts with a general class of FIR filters called Order Statistic (OS) filters. In contrast to linear filters, where temporal ordering of the samples is preserved, OS filters base their operation on the ranking of the samples within the filter window. The data are ranked by their summary statistics, such as their mean or variance, rather than by their temporal position.

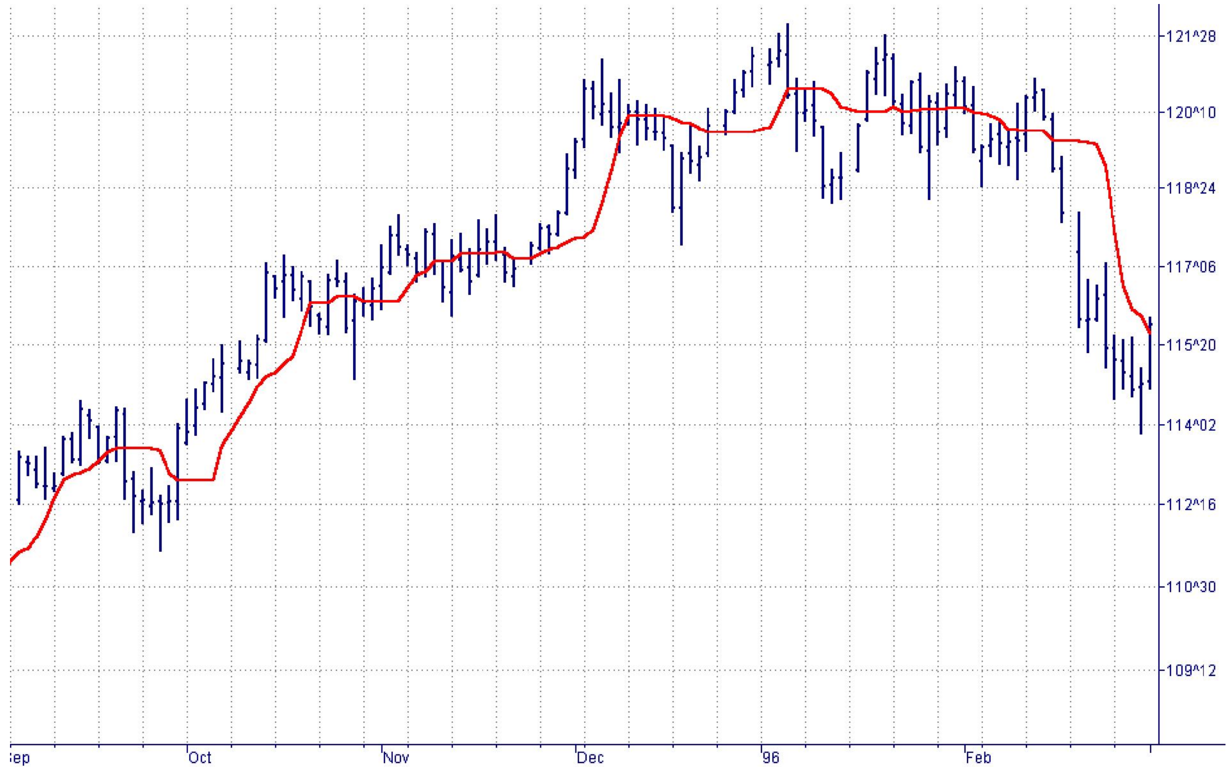
Among OS filters, the median filter is the best known. It is used in video circuits to sharpen the edges of images and to remove impulsive noise. In a median filter, the output is the median value of all the data values within the observation window. As opposed to an averaging filter, the median filter simply discards all data except the median value. In this way, impulsive noise spikes and extreme price data are eliminated rather than being included in the average. The median value can fall at the first sample in the data window, at the last sample, or anywhere in between. Thus, temporal characteristics are lost. For example if the data inputs to a five tap FIR filter are sequentially [3 4 3 3 9], the median value is 4. This median is neither the average value (which is 4.4) nor the value at the center of the filter. In this case the big data spike value of 9 is ignored. As another example, if the data inputs were [3 3 3 8 9] the median value is 8, as opposed to the average value of 5.2. This example shows the median filter can follow fast changes in data rapidly.

The median filter tends to smooth out short-term variations that lead to whipsaw trades with linear filters. On the other hand, the lag of a median filter in response to a sharp and sustained price movement is substantial – it necessarily is about half the filter window width. The response of a median filter that has a 10 bar window width is shown in Figure 1. Note that the filter did not respond to small price movements in October/November nor in January/February, which possibly could have eliminated several potential whipsaw trades that would have been produced by linear filters.

Finding the median is a simple sorting problem. We only need to list the data samples within the filter width in the order of the their amplitudes and pick the center value. Median filters can be smoothed with an Exponential Moving Average to make them more presentable and easier to read.

Like OS filters, Ehlers filters are robust. Additionally, they also exploit both the rank-order and the temporal characteristics of the data. The generalized Ehlers filter can be oriented to any statistic of your choice, making the coefficients extremely easy to calculate. The most obvious statistic to use is price momentum because this data enables the nonlinear Ehlers filter to rapidly follow price changes (as it enabled the KAMA IIR filter to do the same). The range of statistic used is virtually limitless. For example, the Ehlers filter could be nonlinear with respect to acceleration (the rate change of momentum), signal-to-noise ratio, volume, money flow (delta price times volume), and so on. Even other indicators, such as Stochastic or RSI can be used as a statistic. This will become more apparent after I explain the calculating procedure.

Figure 1. Response of a 10 Bar Median Filter



The Ehlers filter has a formulation similar to that of the FIR filter. If “y” is the filter output and “x_i” is the i^{th} input across a filter window width “n”, then the equation is:

$y = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 + \ldots\ldots + c_nx_n$

The c’s are the coefficients that contain the statistic in which you are interested. As a simple example, a weighted moving average uses a statistic inversely proportional to the data sample position within the averaging window. In TradeStation notation, the equation for such a filter would be written as:

y = (4*Price + 3*Price[1] + 2*Price[2] + Price[3]) / 10

Note that a weighted moving average has each coefficient normalized to the sum of the coefficients. This normalization keeps the average scaled that the same as the price on your charts. For example, if all 5 prices within the filter are the same, then the filter output necessarily is the same as any of those 5 prices. As a more general example, if you are interested in the 5 bar momentum, each coefficient would be, in EasyLanguage notation:

Price[count] – Price[count + 5]

Where count is the position within the filter window

In this way, the coefficients are ordered according to their size within the window. For example, c_3 could possibly have the largest momentum and c_1 could be the next largest momentum, and their temporal location within the filter is retained. Unity gain of the filter is retained by normalizing all of the coefficients to their sum, just as we did for the weighted moving average. So, the complete formal description of the Ehlers filter is:

$$y = \frac{\sum_{i=0}^{n-1} c_i x_i}{\sum_{i=0}^{n-1} c_i}$$

The statistic used in the Ehlers filters should be detrended for maximum effectiveness. If we do not detrend the statistic, each of the coefficients will have a large common term relative to any differences there may be between them. For example, if a 5 tap filter has the statistic as [-1 –2 0 2 1], the statistic has no common term, and there is a large percentage change between positions within the filter. However, if the statistic is [99 98 100 102 101], the coefficients have a common term of 100 although the difference between coefficients is the same as in the first example. In this latter case there is only a small percentage difference between the coefficients and the Ehlers filter would have performance virtually indistinguishable from a simple moving average.

The EasyLanguage code for the Ehlers filter is given in Figure 2 for the particular example of a 5 bar momentum.

Figure 2. EasyLanguage Code to Compute Ehlers Filters

```
Inputs: Price((H+L)/2),
        Length(15);

Vars:   count(0),
        SumCoef(0),
        Num(0),
        Filt(0);

Array:  Coef[25](0);

{Coefficients can be computed using any statistic of choice ----
    ----- a 5 bar momentum is used as an example}

For count = 0 to Length - 1 begin
      Coef[count] = AbsValue(Price[count] - Price[Count + 5]);
end;

{Sum across the numerator and across all coefficients}
Num = 0;
SumCoef =0;
For count = 0 to Length -1 begin
      Num = Num + Coef[count]*Price[count];
      SumCoef = SumCoef + Coef[count];
end;
Filt = Num / SumCoef;

Plot1(Filt, "Ehlers");
```

The example filter has fifteen coefficients, although the array of coefficients is dimensioned to 25 to allow experimentation using a longer filter (if a filter longer than 25 samples is desired, the dimension of the Array must be increased accordingly). In the first calculation we find each coefficient in the filter as the five bar momentum. The next computation is to sum the numerator as the product of each coefficient and the price (the x’s in the general equation) at the corresponding sample, and sums the coefficients alone. Finally, the filter is completed by taking the ratio of the numerator to the coefficient sum. The performance of this filter is shown in Figure 3.

Figure 3 illustrates the fact that the momentum-derived Ehlers filter clearly responds quickly to rapid price movements while rejecting minor price movements to a greater degree. This kind of filter can be used to quickly respond to changes in trend direction without producing the whipsaws that are so prevalent when linear filters are employed. The Ehlers filter can be made to be very aggressive by squaring each coefficient.

The greater flexibility of Ehlers filters opens up whole new avenues of technical analysis research. For example, the statistic can be some tangible parameter of market activity such as money flow or volume. On the other hand, more arcane parameters such as Signal to Noise Ratio can be used as the statistic. In this case, the coefficients where the Signal to Noise Ratio is the greatest would have the largest weight, discounting the price data values where the Signal to Noise Ratio is less. Also, Ehlers filters can be adaptive. For example, the length of the 5 Bar Momentum Ehlers filter in our example could be adaptive to the length of the measured cycle period. Such a filter would be both adaptive and nonlinear.

The flexibility and adaptability of the Ehlers filter is demonstrated in Figure 4, where the statistic used is the difference between the current price and the previously calculated value of the filter. This version has some aspects of an IIR filter.

Figure 3. Performance of a 15 Bar Ehlers Filter using a 5 Bar Momentum Compared to the Performance of a 15 Bar SMA

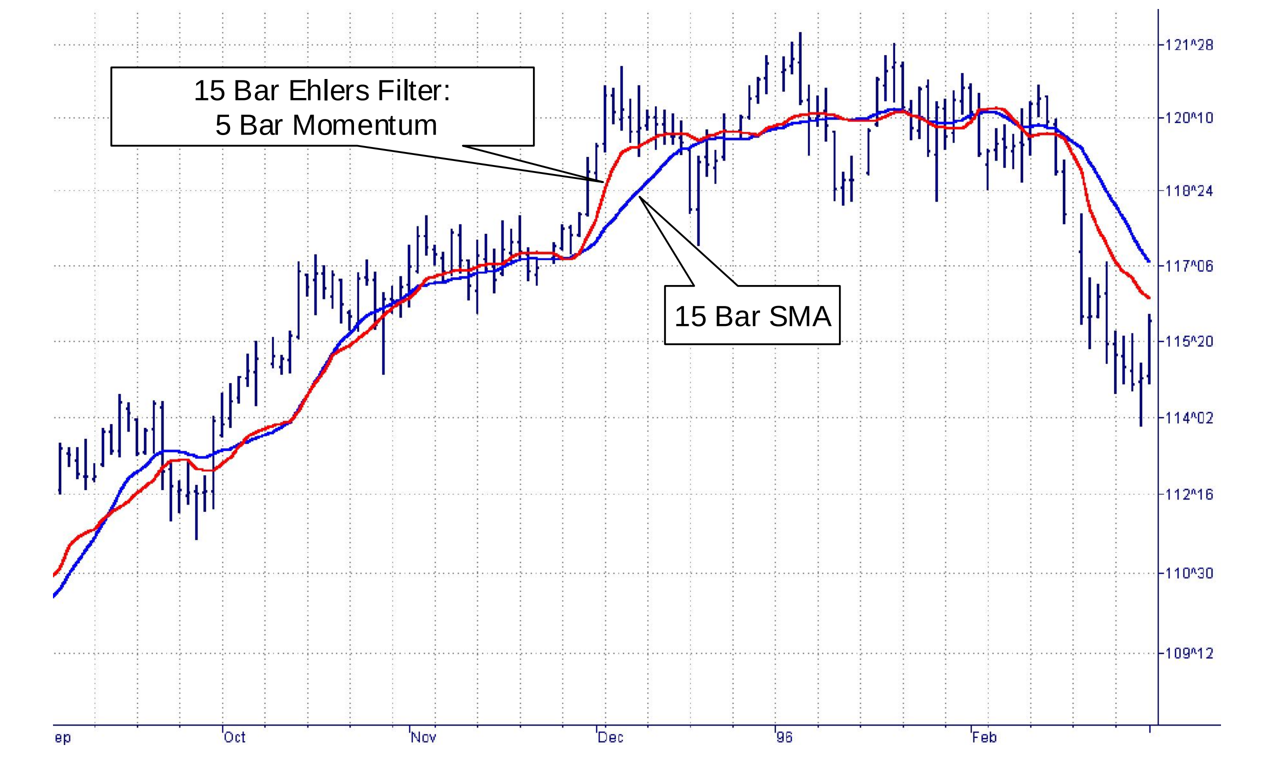
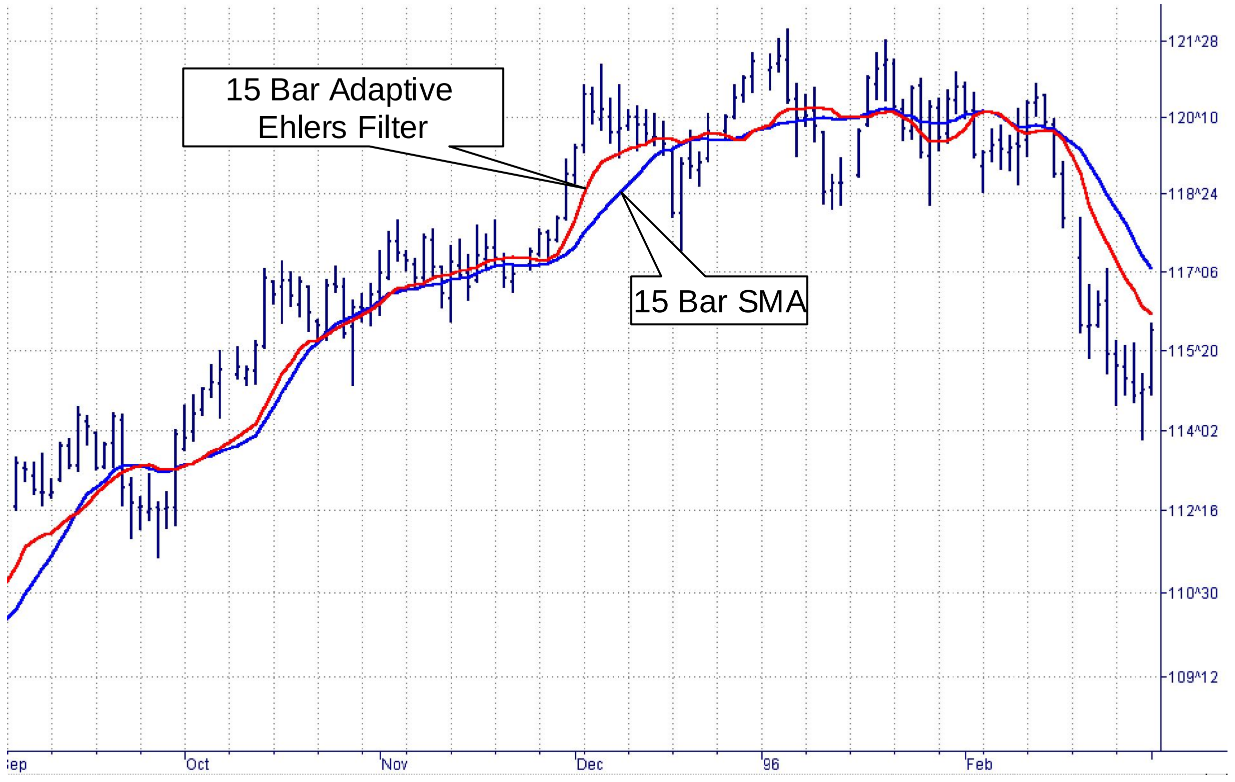
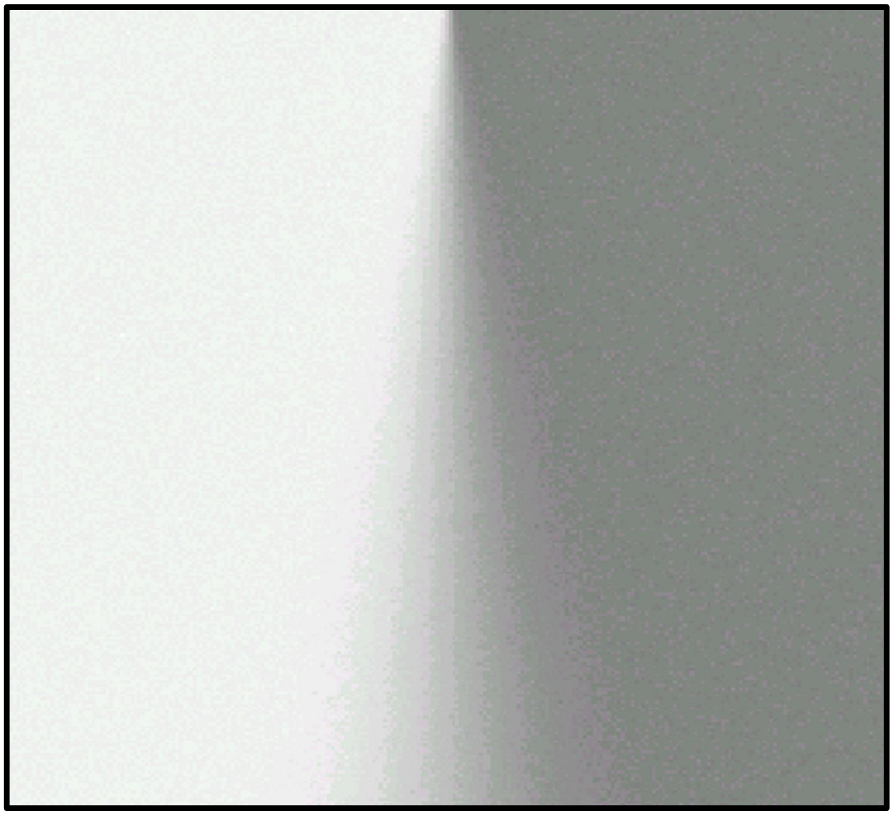


Figure 4. Performance of an Adaptive 15 Bar Ehlers Filter Compared to the Performance of a 15 Bar SMA



Regardless of the flexibility of the Ehlers filter, it is useful to step back and reflect on the motivation for deriving this filter type. By so doing, we may discover an optimum solution for the calculation of the coefficients. We know market data is most often nonstationary. We also know that we want to follow the sharp and sustained movements of price as closely as possible. This led video engineers to use the median filter as an edge detector. But not all edges are the same. We can visualize the sharpness of edges in Figure 5 by imaging looking down on this figure as we would on a piece of paper, illuminated from above our left shoulder, and hanging over the edge of a desk. The edge at the top of Figure 5 is very sharp, as if the paper were creased. Continuing down Figure 5 the light diffusion is more dispersed, giving the illusion that the edge becomes more rounded. In fact, the shading of Figure 5 was generated by a Gaussian function whose standard deviation increased from top to bottom in the figure.

Figure 5. Visualizing the Sharpness of an Edge



If we consider the gray shading levels in Figure 5 as distances, we have a way of computing filter coefficients in terms of sharpness of the edge. White is the maximum distance in one direction from the median gray and black is the maximum distance in the other direction. Thus, distance is a measure of departure from the edge, taking into account the edge sharpness.

Transitioning to price charts, the difference in prices can be imagined as a distance. Recalling the Pythagorean theorem (where the length of the hypotenuse of a triangle is equal to the sum of the squares of the lengths of the other two sides), we can apply it to our needs and say that a generalized length at any data sample is the square root of the sum of the squares of the price difference between that price and each of the prices back for the length of the filter window. The sum of the distances squared at each data point are the coefficients of the Ehlers filter. Suppose the last 10 data values were [1 1 1 1 1 1 2 3 4 5]. The coefficients of the Ehlers filter would then be calculated as:

$C_1 = 1 + 2^2 + 3^2 + 4^2 + 4^2 = 46$
$C_2 = 1 + 2^2 + 3^2 + 3^2 + 3^2 = 32$
$C_3 = 1 + 2^2 + 2^2 + 2^2 + 2^2 = 17$
$C_4 = 1 + 1 +1 + 1 + 1 = 5$
$C_5 = 0$

The calculation of the distance-like coefficients are perhaps best understood with reference to the EasyLanguage code for the filter in Figure 6. If the prices across the filter observation window are the same, then the coefficients of the filter are all the same, and we have the equivalent of a SMA. On the other hand, if the prices shift rapidly, the distances from the increased price points increase, and higher weights are given to these filter coefficients. The performance of the distance coefficient Ehlers filter is shown in Figure 7.

Figure 6. EasyLanguage Code for the Distance Coefficient Ehlers Filter

```
Inputs:     Price((H+L)/2),
            Length(15);

Vars:       count(0),
            LookBack(0),
            SumCoef(0),
            Num(0),
            Filt(0);

Array:      Coef[25](0),
            Distance2[25](0);

For count = 0 to Length - 1 begin
      Distance2[count] = 0;
      For LookBack = 1 to Length - 1 begin
            Distance2[count] = Distance2[count] + (Price[count] – Price[count
+ LookBack])*(Price[count] - Price[count + LookBack]);
      end;
      Coef[count] = Distance2[count];
end;
Num = 0;
SumCoef =0;
For count = 0 to Length -1 begin
      Num = Num + Coef[count]*Price[count];
      SumCoef = SumCoef + Coef[count];
end;
If SumCoef <> 0 then Filt = Num / SumCoef;

Plot1(Filt, "Ehlers");
```

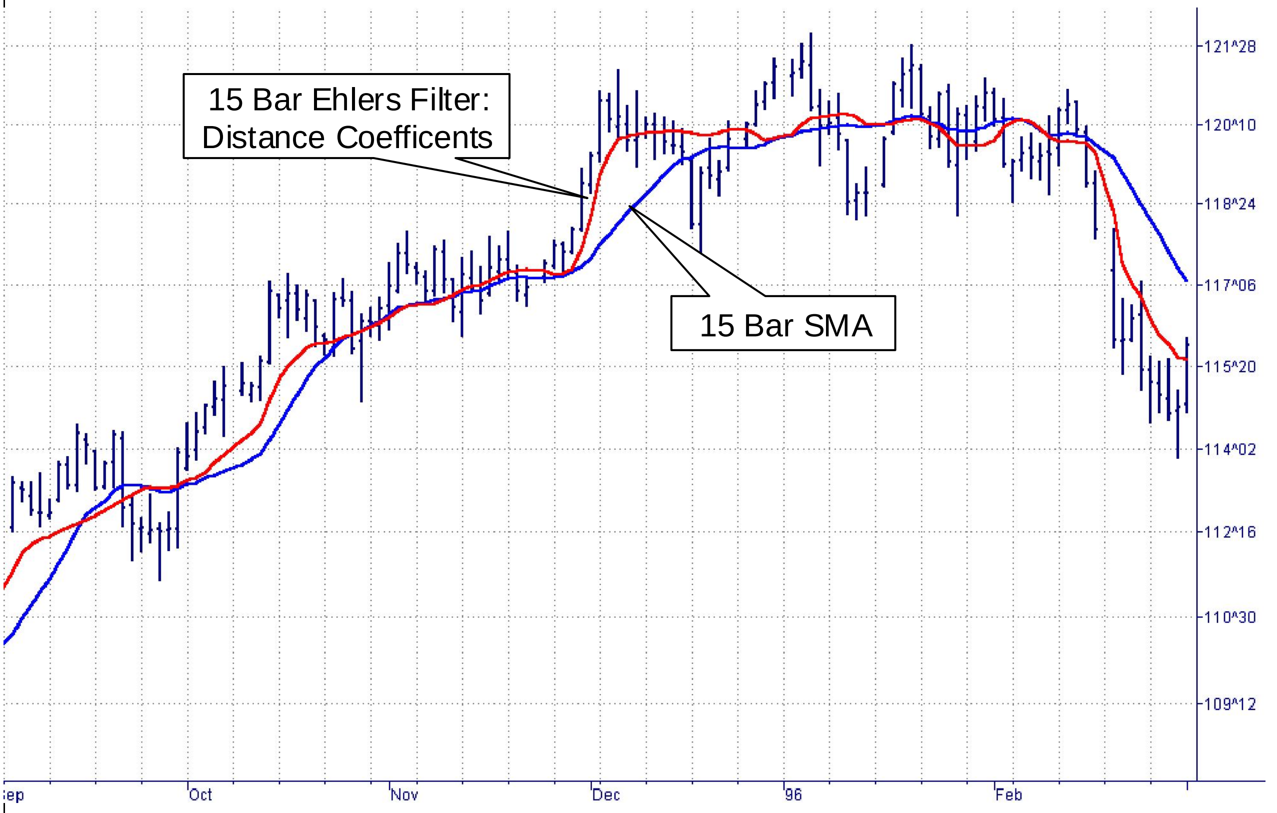
The filter coefficients can be made to be even more nonlinear than calculated in Figure 6. For example, the distance can be cubed or raised to the fourth power (by squaring the squared distance). A reciprocal Gaussian response is an even more nonlinear function of distance that we can use to calculate the filter coefficients. These more nonlinear responses follow the edges in price movement more aggressively. However, the very fact that they are so nonlinear removes much of the gray area in the response.

The most nonlinear calculations produce results that are not discernable from median filters. The coefficients become black and white, so there is very little middle ground gray area. The focus of my current research is to identify the onset of the price shift more accurately. The currently calculated distance functions are related to the change of price. In calculus terms, this is the first derivative. The shift of the rate change of price is the ideal identifier for the impending price move. In calculus terms, we can use the maximum of the second derivative to pinpoint the onset of the price change. The challenge is how to translate the second derivative into filter coefficients without introducing so much noise that the filter response is unusable.

The opportunities to use Ehlers filters in technical analysis are limitless. I am sure whole books will be devoted to cataloging the various statistics and applications where they work the best. In the meantime, you will have had the opportunity of exploiting them for your own fun and profit.

John Ehlers is the president of MESA Software and is a frequent contributor to Stocks & Commodities. This article is adapted from his forthcoming book, ROCKET SCIENCE FOR TRADERS, published by John Wiley & Sons, with availability later this Spring.

Figure 7 Performance of the Distance Coefficient Ehlers Filter



15 Bar Ehlers Filter: Distance Coefficents

15 Bar SMA

SIDEBAR

KAUFMAN'S ADAPTIVE MOVING AVERAGE (KAMA)[1]

Kaufman's Adaptive Moving Average (KAMA) is based on the concept that a noisy market requires a slower trend than one with less noise. The basic principle is that the trendline must lag further behind the price in a relatively noisy market to avoid being penetrated by the price. The moving average can speed up when the prices move consistently in one direction. According to Perry Kaufman, who invented the system, KAMA is intended to use the fastest trend possible, based on the smallest calculation period for the existing market conditions. It does this by changing the alpha of the EMA with each new sample. The equation for KAMA is:

KAMA = S*Price + (1 – S)*KAMA[1]

Where S = Smoothing factor

This is exactly the same equation that we use for the EMA except the variable S replaces the alpha constant of the EMA.

The equation for the Smoothing factor involves two boundaries and an Efficiency Ratio.

$S = (E^*(\text{fastest} – \text{slowest}) + \text{slowest})^2$

Fastest means the alpha of the shortest period boundary. Slowest means the alpha of the longest period boundary. The suggested period boundaries are 2 and 30 bars. In this case the two alphas are calculated to be:

Fastest = 2/(2+1) = .6667

Slowest = 2/(30+1) = .0645

Simplifying the equation for the Smoothing factor, we get:

$S = (.6022^*E + .0645)^2$

The efficiency ratio (E) is the absolute value of the difference of price across the calculation span divided by the sum of the absolute value of the individual price differences across the calculation span. The equation for E is

$$E = \frac{\left|\mathrm{Pr}\,ice - \mathrm{Pr}\,ice[N]\ \right|}{\sum_{i=0}^{N}\left|\mathrm{Pr}\,ice[i] - \mathrm{Pr}\,ice[i+1]\ \right|}$$

The default value for N is 10. However testing to find the best length is suggested.

[1] Perry J. Kaufman, "Trading Systems and Methods, Third Edition", John Wiley & Sons, pp 436-438

VARIABLE INDEX DYNAMIC AVERAGE (VIDYA)[2]

Variable Index Dynamic Average (VIDYA) uses a pivotal smoothing constant which is fixed. The suggested value of this constant is 0.2, corresponding to the alpha of a 9 day EMA. The equation for VIDYA is

VIDYA = 0.2*k*Close + (1 - .2*k)*VIDYA[1]

Again, this is exactly the same equation as an EMA except the relative volatility term, k, has been included to introduce the nonlinearity. The volatility term is the ratio of the standard deviation of Closes over the last “n” days to the standard deviation of Closes over the last “m” days, where m is greater than n. Suggested values are n=9 and m=30.

[2] Tushar S. Chande and Stanley Kroll, “The New Technical Trader”, John Wiley & Sons, New York, 1994