

FRAMA – Fractal Adaptive Moving Average

By John Ehlers

The objective of using filters is to separate the desired signals from the undesired signals (or noise). The practical application of moving averages often involves a tradeoff between the amount of smoothness required and the amount of lag that can be tolerated. Moving averages have this problem because the price data is not stationary, and may have different bandwidths over different time intervals. Various momentum-adaptive filtering techniques¹² have been developed to take advantage of the nonstationary structure of prices. Adaptive filters have also been developed based on price statistics³ and the cyclic content of the price data⁴. In this article I will describe a different class of filters that monitor a different measure of temporal nonstationarity and alters their bandwidth in response to this measure.

There is no argument that market prices are fractal. Fractal shapes are self-similar because they tend to have the same roughness and sparseness regardless of the magnification used to view them. If you remove the labels from a 5 minute chart, a daily chart, and a weekly chart you would have difficulty telling them apart. This is the characteristic that makes them fractal. The self-similarity can be defined by the fractal dimension that describes the sparseness at all magnification levels.

To determine the fractal dimension of a generalized pattern, we cover the pattern with a number “N” of small objects of several various sizes “s”. The relationship of the number of objects in two sets of sizes is:

$$\frac{N2}{N1} = \left(\frac{s1}{s2} \right)^D$$
$$D = \frac{\log(N2/N1)}{\log(s1/s2)}$$

As an example of computing the fractal dimension, we start with a line segment that is 10 meters long. We chose the two small dimensions as s1 = 1 meter and s2 = 0.1 meter. Placing boxes on the line, we can fit ten 1 meter on a side of the 10 meter line segment. Therefore N1 = 10. Similarly, we can fit one hundred boxes that are 0.1 meters on a side on the 10 meter line. Therefore N2 = 100. The fractal dimension of the line computes to be:

¹ KAMA, Kaufman, Perry J., “Trading Systems and Methods”, 3rd ed., New York, John Wiley & Sons, 1998

² Ehlers Filters, Ehlers, John F. “Cybernetic Analysis for Stocks and Futures”, New York, John Wiley & Sons, 2004

³ VIDYA, Chande, Tushar S. and Stanley Kroll, “The New Technical Trader”, New York, John Wiley & Sons, 1994

⁴ MAMA, Ehlers, John F., “Rocket Science for Traders”, New York, John Wiley & Sons, 2001

$$D = \frac{\log(100/10)}{\log(1/0.1)} = 1.0$$

As a second example, we will use the pattern as a square that is 10 meters on a side. Retaining the same sizes of our small square objects as 1 meter and 0.1 meter respectively, we get $N1 = 100$ and $N2 = 10,000$. When the square is our pattern, the fractal dimension therefore computes to be:

$$D = \frac{\log(10000/100)}{\log(1/0.1)} = 2.0$$

Natural fractals, such as a shoreline, lack the true regularity of an algorithmic structure but are self-similar in a statistical sense. Thus, in order to determine the fractal dimension of these natural shapes, we must average the measured fractal dimension made over different scales.

We could measure the fractal dimension of prices by covering the curve with a series of small boxes. This is burdensome task, but if we note that the price samples are uniformly spaced, we note that the box count is approximately the average slope of the curve. Therefore, we can estimate the box count as the highest price during an interval minus the lowest price during that interval, divided by the length of the interval itself. The equation for the number of boxes is then:

$$N1 = \frac{\text{Highest Price} - \text{Lowest Price}}{\text{Length}}$$

We compute the fractal dimension by computing “N” over two equal intervals to get the averaging over each interval. Box1 covers the period from 0 to T bars ago. Box2 covers the period from T to 2T bars ago. Therefore, $N1 = (\text{HighestPrice} - \text{LowestPrice})$ over the interval from 0 to T, divided by T. Similarly, $N2 = (\text{HighestPrice} - \text{LowestPrice})$ over the interval from T to 2T, divided by T. We also define a $N3 = (\text{HighestPrice} - \text{LowestPrice})$ over the entire interval from 0 to 2T, divided by 2T. Since we are looking backwards in time, the slope computation of the fractal dimension is:

$$D = \frac{\log(N1 + N2) - \log(N3)}{\log(2)}$$

The fractal dimension varies over the range from D=1 to D=2. I use the fractal dimension to dynamically change the alpha of an exponential moving average. Since the prices are log-normal, it seems reasonable to use an exponential function to relate the fractal dimension to alpha. I chose the relationship as:

$$\alpha = \exp(-4.6 * (D - 1))$$

When D=1, the exponent is zero – which means that $\alpha = 1$. When alpha is one the output of the exponential moving average is equal to the input. That's about as fast as anyone can make an average. On the other hand $\alpha = 0.01$ when D=2. This is a very slow moving average, analogous to a 200 day simple moving average. Thus, our FRAMA swings between being a fast moving average when D=1 and a very slow moving average when D=2. This adaptive structure rapidly follows major changes in price and slowly changes when the prices are in a congestion zone.

The EasyLanguage code to compute FRAMA is given in Figure1. This code directly follows the development process I described in the text. The Length input is the entire period to compute N3. Since this period is divided into two equal segments for N1 and N2, the length must be an even number. For the more adventurous reader, it might be instructive to plot the fractal dimension (the variable Dimen) as a standalone indicator to see if it creates some helpful signals.

Figure 1. FRAMA EasyLanguage Code

```
{
    *****
    FRAMA
    By John Ehlers
    ***** }

Inputs: Price((H+L)/2),
        N(16);
        {N must be an even number}

Vars:  count(0),
        N1(0),
        N2(0),
        N3(0),
        HH(0),
        LL(0),
        Dimen(0),
        alpha(0),
        Filt(0);

N3 = (Highest(High, N) - Lowest(Low, N)) / N;
```

```

HH = High;
LL = Low;
For count = 0 to N / 2 - 1 begin
    If High[count] > HH then HH = High[count];
    If Low[count] < LL then LL = Low[count];
End;
N1 = (HH - LL) / (N / 2);

HH = High[N / 2];
LL = Low[N / 2];
For count = N/2 to N - 1 begin
    If High[count] > HH then HH = High[count];
    If Low[count] < LL then LL = Low[count];
End;
N2 = (HH - LL) / (N / 2);

If N1 > 0 and N2 > 0 and N3 > 0 then Dimen = (Log(N1 + N2) - Log(N3)) / Log(2);

alpha = ExpValue(-4.6*(Dimen - 1));
If alpha < .01 then alpha = .01;
If alpha > 1 then alpha = 1;

Filt = alpha*Price + (1 - alpha)*Filt[1];
If CurrentBar < N + 1 then Filt = Price;

Plot1(Filt);

```

As with any moving average, we are forced to compromise between responsiveness and smoothness. For that reason, the Length parameter is an input that can easily be changed. Figure 2 shows the adaptivity and the smoothness of FRAMA.

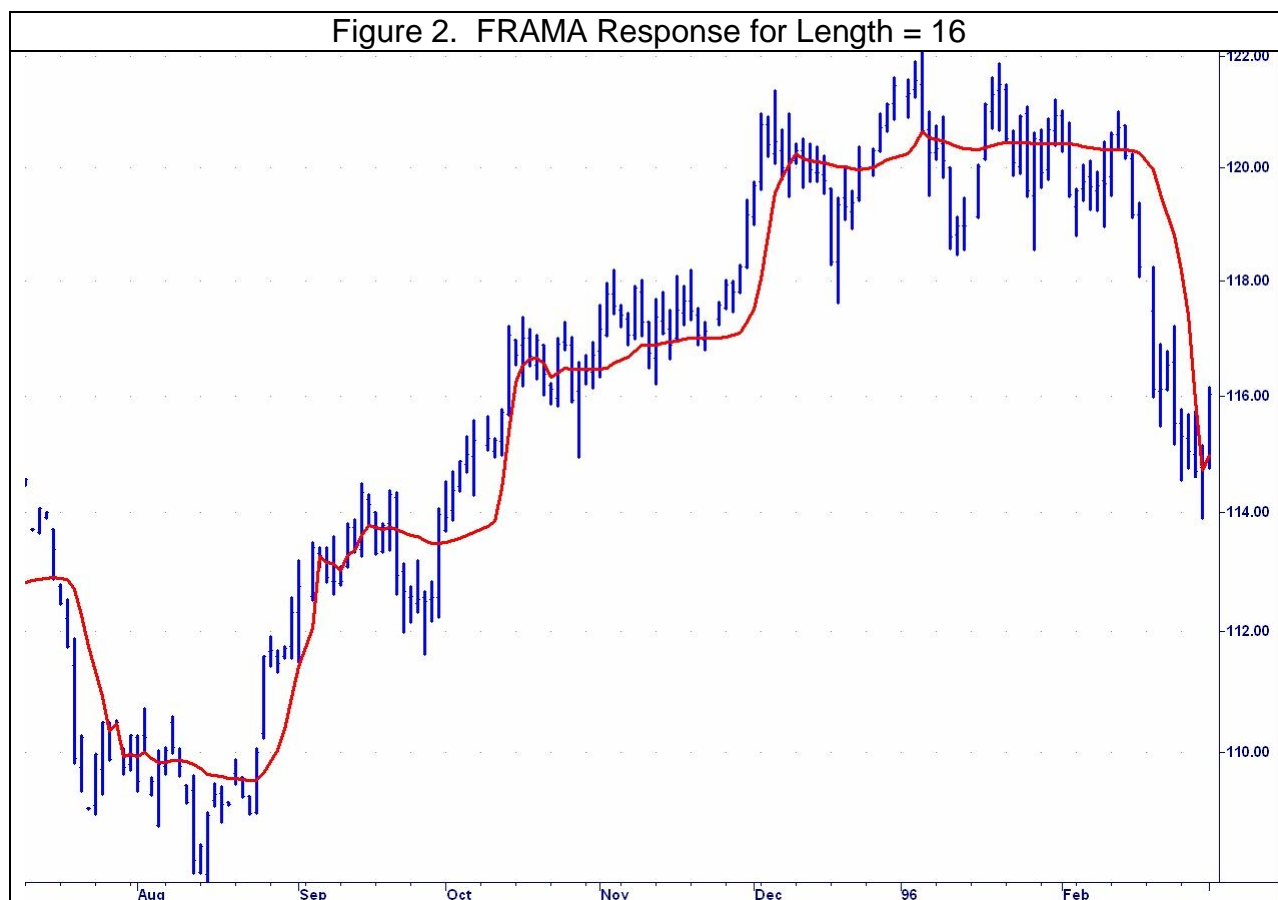


Figure 2 shows that FRAMA can be a valuable weapon in your arsenal of technical indicators. It rapidly follows significant changes in price but becomes very flat in congestion zones so that bad whipsaw trades can be eliminated.