

THE ULTIMATE SMOOTHER

By John Ehlers

A smoother is a Low Pass filter that passes the low frequency components of the input data spectrum essentially unchanged and rejects or attenuates the high frequency components in the data. This rejection of the high frequency components produces an output waveform that is smoother than the input waveform. Early in my engineering career I designed filters using real inductors and real capacitors. This experience enabled me to invent the SuperSmoother¹² as well as the UlitimateSmoother described later in this article.

SUPERSMOOTHER

The SuperSmoother digital filter was translated from an analog filter having a maximally flat Butterworth low pass response with a reduced lag. The EasyLanguage code for the SuperSmoother is repeated in Code Listing 1. Here, it is written as a function so that it can be used in indicators and strategies as easily as a moving average.

Code Listing 1. SUPERSMOOTHER FUNCTION

```
{
    SuperSmoother Function
    (C) 2004-2024 John F. Ehlers
}

Inputs:
    Price(numericseries),
    Period(numericinteger);

Vars:
    a1(0),
    b1(0),
    c1(0),
    c2(0),
    c3(0);

a1 = expvalue(-1.414*3.14159 / Period);
b1 = 2*a1*Cosine(1.414*180 / Period);
c2 = b1;
c3 = -a1*a1;
c1 = 1 - c2 - c3;
```

¹ John Ehlers, CYBERNETIC ANALYSIS FOR STOCKS AND FUTURES, John Wiley & Sons, 2004

² John Ehlers, "Predictive and Successful Indicators", TASC, January 2014

```
If CurrentBar >= 4 Then $SuperSmoother = c1*(Price + Price[1]) / 2 +  
c2*$SuperSmoother[1] + c3*$SuperSmoother[2];  
If Currentbar < 4 Then $SuperSmoother = Price;
```

The SuperSmoother is a second order IIR (acronym for Infinite Impulse Response) filter, meaning that it uses two previous calculations of the filter output in the current calculation of the filter response. The defining parameter of the SuperSmoother is the critical period. Critical period is the wavelength that divides the pass band and the reject band. Conceptually, think of it being the wavelength where all longer wavelengths are passed unaffected to the output and all shorter wavelengths are completely rejected at the output. Actually, there is a more gradual transition from the pass band to the stop band. The sharpness of this transition, to a greater degree, depends on the order of the IIR filter. For brevity, critical period is the input "Period" in Code Listing 1.

An Exponential Moving Average (EMA) is a first order IIR filter, using only one previous calculation. It is a smoother described by the equation:

$$\text{EMA} = \alpha * \text{Price} + (1 - \alpha) * \text{EMA}[1]$$

Where EMA[1] means the value of EMA one bar ago

The performance of an EMA can be compared to the performance of a SuperSmoother by letting alpha be equal to 3 divided by Period. The code to plot the comparison of these two smoothers is given in Code Listing 2, and the comparison chart is shown in Figure 1.

Code Listing 2. Plot SuperSmoother and EMA

```
Inputs:  
    Length(20);  
  
Vars:  
    SS(0),  
    EMA(0),  
    aa(0);  
  
SS = $SuperSmoother(Close, Length);  
  
aa = 3 / Length;  
EMA = aa*Close + (1 - aa)*EMA[1];  
  
Plot1(SS, "", red, 2, 2);  
Plot2(EMA, "", blue, 2, 2);
```



Figure 1. SuperSmoother Has a Better Response Than an EMA for Equivalent Lag

It is obvious that the second order SuperSmoother (in red) has superior smoothing than a first order EMA having an equivalent critical period (in blue). It is also obvious that both filters have a lag compared to the input price data. Lag of smoothing filters is a major bane for technical traders. It doesn't help to get the right answer if it comes too late.

It is possible to design higher order smoothing filters. Higher order filters sharpen the transition from the pass band to the stop band. However, the penalty is that such filters incur even more lag. Therefore, these higher order filters are seldom employed in trading. In addition, the calculation of higher order filters causes floating point errors on many trading platforms. A reasonable approximation to the response of a fourth order filter without getting a floating-point error is to first filter the data with a SuperSmoother and then filter that result with a SuperSmoother again.

HIGHPASS FILTER

Low pass filters are not the only kind of filters that can be employed. If the roles below and above the critical period are reversed, a High Pass filter is created. That is, the low frequency components in the input data are rejected at the filter output and the high frequency components are passed to the output unattenuated. The EasyLanguage code for a High Pass filter function is given in Code Listing 3.

Code Listing 3. EasyLanguage High Pass Filter Function
{ Highpass Function

```
(C) 2004-2024 John F. Ehlers
```

```
}
```

```
Inputs:
```

```
Price(numericseries),  
Period(numericimple);
```

```
Vars:
```

```
a1(0),  
b1(0),  
c1(0),  
c2(0),  
c3(0);
```

```
a1 = expvalue(-1.414*3.14159 / Period);  
b1 = 2*a1*Cosine(1.414*180 / Period);  
c2 = b1;  
c3 = -a1*a1;  
c1 = (1 + c2 - c3) / 4;
```

```
If CurrentBar >= 4 Then $HighPass = c1*(Price - 2*Price[1] + Price[2]) +  
c2*$HighPass[1] + c3*$HighPass[2];  
If Currentbar < 4 Then $HighPass = 0;
```

BANDPASS FILTER

Another kind of filter that can be designed is a Band Pass filter. With this type of filter there is a band of periods that are passed basically unattenuated to the output, periods longer than the lower critical period are rejected, and periods above the upper critical period are also rejected. I have designed several second order Band Pass filters, but I discourage their use because there is a temptation to use a relatively narrow bandwidth with such filters to get an output that looks similar to a Sine Wave. The sine wave appearance results from the filter passes only a narrow portion of the data spectrum to the output. When the bandwidth is on the order of twenty five percent of the center period or less, the phase shift across the pass band is nearly continuous. The phase response, including the transition bands, approaches 180 degrees. So, using such a filter can give fantastic in-phase response for a while. But if the character of the data shifts just a little, the output waveform can give a dead wrong trading signal due to the 180 degree phase shift. Recognizing the shift in the input data is nearly impossible in real time.

A better way to create a Band Pass filter response is to use a SuperSmoother with the Period set to the Upper Critical Period of the Pass Band and a High Pass filter with the

Period set to the Lower Critical Period of the Pass Band. The separation between the Lower Critical Period and the Upper Critical Period should be at least one octave. This way, the relative phase shift across the pass band is nearly consistent. Code Listing 4 is an example of such a Band Pass filter. The pass band is set to be the octave between a 15 bar cycle period and a 30 bar cycle period. These inputs can be adjusted to obtain the best results for a given set of data. The Band Pass filter output is an oscillator-style indicator and can be used directly as an indicator. An example of the filter response is shown in Figure 2.

Code Listing 4. EasyLanguage Band Pass Filter

```
{
    BandPass Filter
}

Inputs:
    LowerPeriod(30),
    UpperPeriod(15);

Vars:
    HP(0),
    BP(0);

HP = $HighPass(Close, LowerPeriod);
BP = $SuperSmoother(HP, UpperPeriod);

Plot1(BP, "", blue, 4, 4);
Plot2(0, "", black, 2, 2);
```



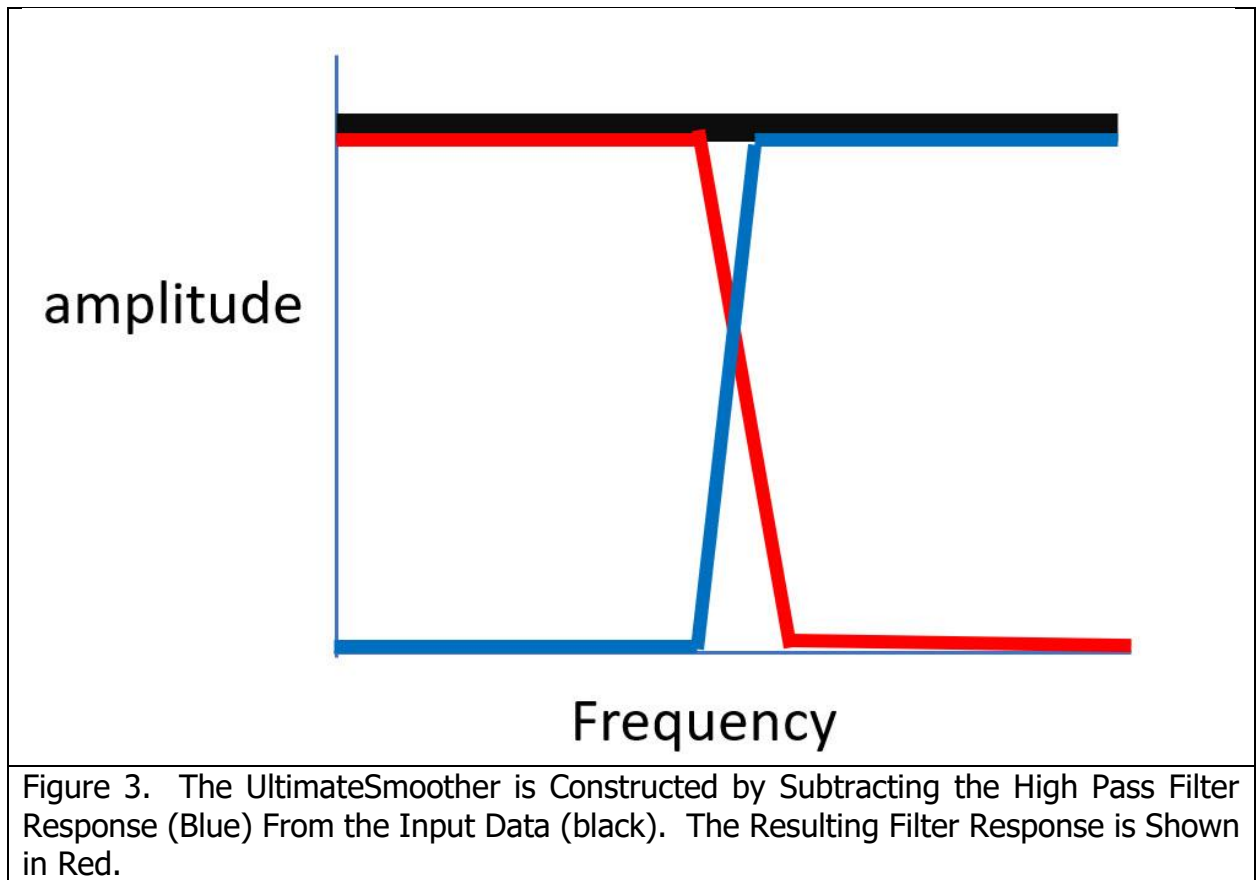
This BandPass filter is superior to a second order BandPass filter because it provides sharper transitions between the pass band and the stop bands. Think of it this way: The second order BandPass applies one order to the upper filter edge and one order to the lower filter edge, whereas this BandPass filter applies two orders to both the upper and lower band edges.

ULTIMATE SMOOTHER

Like the BandPass filter, the UltimateSmoother is comprised of two component filters. Smoothing filters always involve lag in their output, and lag is to be avoided if possible. From my analog filter design experience, I know that lower frequency filters require larger inductors and capacitors, and the lag results from the increased energy required to build up the electric and magnetic fields in these components. Think in terms of big woofer speakers for low frequency sound and tiny tweeters that produce high frequency sound. That concept is lost with digital filters, where the filters are just code. But the principle is the same. You know a moving average becomes smoother the longer you make the average. That is the same thing as putting more energy into the filter. You also know that longer moving averages have more lag.

The UltimateSmoother conceptually has zero lag in the Pass Band and has minimum lag in the transition band because only a high frequency filter is involved. The idea of the UltimateSmoother is described with reference to the schematic in Figure 3. The input data is mathematically described as an All Pass filter, shown in black. The response of a High Pass filter is shown in blue. The Ultimate Smoother response is a result of subtracting the High Pass response from the All Pass response, and is shown in red. At

the very low frequencies the High Pass filter has virtually no amplitude, and so the result of the subtraction is that the UltimateSmoother output is the same as the input data in terms of both amplitude and lag. On the other hand, the response of the High Pass filter is almost the same as the input data and therefore the filtering is accomplished by cancellation.



For the mathematically inclined, the subtraction of the High Pass filter from the All Pass filter transfer function is described in terms of Z Transforms as:

$$\text{Transfer} = 1 - \frac{c1*(1 - 2*Z^{-1} + Z^{-2})}{1 - c2*Z^{-1} - c3*Z^{-2}}$$

Putting over a common denominator, the transfer function in closed form is:

$$\text{Transfer} = \frac{(1 - c1) + (2*c1 - c2)*Z^{-1} - (c1 + c3)*Z^{-2}}{1 - c2*Z^{-1} - c3*Z^{-2}}$$

This transfer function is translated to EasyLanguage code in Code Listing 5. Code Listing 5 is the Ultimate Smoother written as a function.

Code Listing 5. EasyLanguage Function for the UltimateSmoother

```
{
    Ultimate Smoother Function
    (C) 2004-2024 John F. Ehlers
}

Inputs:
    Price(numericseries),
    Period(numericimple);

Vars:
    a1(0),
    b1(0),
    c1(0),
    c2(0),
    c3(0),
    US(0);

a1 = expvalue(-1.414*3.14159 / Period);
b1 = 2*a1*Cosine(1.414*180 / Period);
c2 = b1;
c3 = -a1*a1;
c1 = (1 + c2 - c3) / 4;

If CurrentBar >= 4 Then US = (1 - c1)*Price + (2*c1 - c2)*Price[1] - (c1 + c3)*Price[2]
+ c2*US[1] + c3*US[2];
If CurrentBar < 4 Then US = Price;

$UltimateSmoother = US;
```

Writing the EasyLanguage Code for an UltimateSmoother filter is simple, as shown in Code Listing 6.

Code Listing 6. UltimateSmoother Example Filter

```
{
    UltimateSmoother Filter
}

Inputs:
    Period(20);

Vars:
    US(0);
```



```
US = $UltimateSmoother(Close, Period);
```

```
Plot1(US, "", blue, 4, 4);
```

The UltimateSmoother example is plotted in Figure 4. The amazing feature of the UltimateSmoother is that it has zero lag in the Pass Band. The lack of lag in Figure 4 can be compared to the lag of a SuperSmoother and EMA in Figure 1. In all cases the critical period was set to 20 bars. The UltimateSmoother can be applied to any input data, including other indicators.



CONCLUSIONS

1. The UltimateSmoother has zero lag in the Pass Band.
2. The UltimateSmoother is created by subtracting the response of a High Pass filter from the input data.
3. The UltimateSmoother output is not quite as smooth as that of the SuperSmoother because filtering is accomplished by cancellation. Amplitude and phase response of the High Pass filter in its pass band is not exactly the same as that of the input data.
4. The SuperSmoother is recommended for use instead of an EMA. In most cases it can also be used instead of a Simple Moving Average.

5. The best Band Pass filter is created by the serial filtering of a High Pass filter and a SuperSmoother.
6. Band Pass filters should have a bandwidth exceeding an octave.
7. Code for the UltimateSmoother, SuperSmoother, and High Pass filter are provided as functions so they can be called as easily as a moving average.
8. The critical period is the defining parameter for filters. The critical period describes the period that separates the Pass Band from the Stop Band.

John Ehlers is a retired electrical engineer and a retired technical analyst, specializing in the application of DSP to trading. For more information, see www.mesasoftware.com.