

# ZERO LAG (well, almost)

By John Ehlers and Ric Way

All smoothing filters and moving averages have lag. It's a law. The lag is necessary because the smoothing is done using past data. Therefore, the averaging includes the effects of the data several bars ago. In this article we show you how to remove a selected amount of lag from an Exponential Moving Average (EMA). Removing all the lag is not necessarily a good thing because with no lag the indicator would just track out the price you were filtering. That is, the amount of lag removed is a tradeoff with the amount of smoothing you are willing to forgo. We show you the effects of lag removal in an indicator and then use the filter in an effective trading strategy.

An EMA is computed by taking a fraction of the current price and adding to it the quantity  $(1 - \alpha)$  times the previously computed value of the EMA. That fraction is called the smoothing factor and is commonly called  $\alpha$  (alpha), and alpha is always less than 1. The equation for an EMA can be written as:

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

Where EMA[1] is the value of the EMA one bar ago.

Engineers often describe filters in terms of their impulse response. An impulse is where the data has a finite value at one sample and has a zero value at all other samples. Suppose an impulse of data has a value of  $1 / \alpha$ . When the EMA is applied to this impulse the filter output at the end of the first sample is just 1 because there is no previous value in the filter. On the next sample, the input value is zero, and the previous value of 1 is multiplied by  $(1 - \alpha)$ , so the output is  $(1 - \alpha)^1$ . On the third sample there again is no new input and the previous value of  $(1 - \alpha)$  is multiplied by  $(1 - \alpha)$ , so the output is  $(1 - \alpha)^2$ . The process continues with each new data sample, with the result that the filtered output response to the impulse decays as the exponent of the compliment of the smoothing factor. Hence the name, Exponential Moving Average. When alpha is small, the EMA gives a high degree of smoothing and a relatively large amount of lag. When alpha is large (but less than 1) the EMA does very little smoothing and the EMA has very little lag.

Traders should also take note that the sum of the coefficients in the filter,  $\alpha$  and  $(1 - \alpha)$ , sum to 1. This means that if the data input is a constant value, the filtered output will eventually almost reach that constant value also. It is a common coding error to not have the filter coefficients sum to 1. This is why we always write the EMA code to explicitly reflect the smoothing factor and its compliment.

We believe the first relationship between the lag of an EMA and the length of a Simple Moving Average (SMA) was published by Jack Hutson, founder of Stocks & Commodities Magazine.<sup>1</sup> He found that the EMA smoothing factor is related to the length of an SMA as

$$\alpha = 2 / (\text{Length} + 1).$$

---

<sup>1</sup> Jack Hutson, "Stocks & Commodities", (Elizabeth, please help with this – my guess is 1982-1984)

For readers interested, the mathematical derivation of EMA lag related to the length of an SMA can also be found in my book.<sup>2</sup> Since minimization of lag is crucial to the effectiveness of indicators, several authors have devised ways to make the EMA smoothing factor vary with volatility in price.<sup>34</sup> Mark Jurik also has commercially available filters for various platforms that are specifically designed to minimize lag.<sup>5</sup>

The method we describe in this article is fundamentally different from previous approaches. Here, we acknowledge the EMA filter has an error term. That error for the current data sample is just Price – EMA[1]. We introduce this error term into the equation in addition to the value of the new data sample. We further change the amplitude of the error term by multiplying the error by a gain term. We call the new filter EC (for Error Correcting) instead of EMA. So the equation for the EC filter is:

$$EC = a * (Price + gain * (Price - EC[1])) + (1 - a) * EC[1];$$

The equation is simple, but its results are profound. If the gain is zero, the EC becomes just an EMA. If the gain is sufficiently large the error term causes EC to exactly track the price for all practical purposes. That is, there is virtually no lag and virtually no smoothing. Therefore, we seek a value of gain that is a happy medium between tracing out the price and tracing out the EMA. We do this by limiting the maximum amount of allowable gain. If the difference between the price and the previous value of EC is small we do not want a large value of gain. Further, the previous value of EC can be either greater than or less than the current price. Therefore, to properly apply the error correction the gain must swing both positive and negative.

The complete Zero Lag Indicator is described with reference to Figure 1, the computation in EasyLanguage code. There are two user inputs: the length of the equivalent SMA and the Gain Limit. Actually the Gain Limit input is ten times the actual gain used in the computation because the error correcting loop variable increments in steps of 1 and we want the gain to be incremented in steps of 0.1. The smoothing factor alpha and EMA are computed after the variables are declared. Next, a loop searches for the least amount of error, varying gain from the lower Gain Limit to the upper Gain Limit. When a lower error is computed, the values of best gain and least error are recorded. Once the loop has completed its work, the BestGain value is used to compute the Error Corrected (EC) indicator. Then, both the EMA and EC are plotted on the chart.

---

<sup>2</sup> John Ehlers, "Rocket Science for Traders", John Wiley & Sons, New York, pp

<sup>3</sup> Perry J. Kaufman, "Trading Systems and Methods, Third Edition", John Wiley & Sons, pp 436-438

<sup>4</sup> Tushar S. Chande and Stanley Kroll, "The New Technical Trader", John Wiley & Sons, New York, 1994

<sup>5</sup> Mark Jurik, [www.jurikres.com](http://www.jurikres.com)

```
Inputs:  
    Length(20),  
    GainLimit(50);
```

```
Vars:  
    alpha(0),  
    Gain(0),  
    BestGain(0),  
    EC(0),  
    Error(0),  
    LeastError(0),  
    EMA(0);
```

```
alpha = 2 / (Length + 1);  
EMA = alpha*Close + (1 - alpha)*EMA[1];  
LeastError = 1000000;  
For Value1 = -GainLimit to GainLimit Begin  
    Gain = Value1 / 10;  
    EC = alpha*(EMA + Gain*(Close - EC[1])) + (1 - alpha)*EC[1];  
    Error = Close - EC;  
    If AbsValue(Error) < LeastError Then Begin  
        LeastError = AbsValue(Error);  
        BestGain = Gain;  
    End;  
End;  
EC = alpha*(EMA + BestGain*(Close - EC[1])) + (1 - alpha)*EC[1];  
  
Plot1(EC);  
Plot3(EMA);
```

**Figure 1. EasyLanguage Code for the Zero Lag Indicator**

It is good practice to test indicators on theoretical waveforms to see if they behave as expected in controlled circumstances. For example, the Zero Lag indicator response to a step function is shown in Figure 2 where the SMA Length is set to 12 and the Gain Limit is set to 50. In Figure 2 we see that the EC indicator line has a significantly faster rise time and fall time than the EMA. Additionally, there are no transient overshoots usually associated with more complicated indicators.



Figure 2. Zero Lag EC (yellow) and EMA (red) Response to a Step Function

Another theoretical stress test is to see how the indicator responds to a short burst of waves. The purpose of this test is to assess how the indicator will respond to wave-like transients in the data. Figure 3 shows the EC and EMA lines responding to a three cycle burst of waves that are 20 bars long. The Zero Lag Indicator inputs are again set to a SMA Length of 12 and a Gain Limit of 50. Not only are there an absence of significant transients, but the idea of using the EC and EMA line crossovers as entry and exit signals can be formed.



**Figure 3. The Zero Lag Indicator Provides Reliable Crossover Signals**

When we apply the Zero Lag indicator to real market data, using closing prices for example, we know the EC line will faithfully track the closes. So using a high value of Gain Limit would be of no value. Figure 4 shows the Zero Lag Indicator applied to approximately one year of MSFT data. The input settings were Length = 32 and GainLimit = 22. The red line is the EMA and the Yellow Line is the EC Error Corrected line. Since we did not apply the maximum Gain Limit the EC does not faithfully follow the closes, but rather provides a leading indicator relative to the EMA. In general, when the EC is above the EMA the stock is in a bull mode and when the EC is below the EMA the stock is bearish.



Figure 4. The Zero Lag Indicator Provides Crossover Signals for MSFT

The line crossovers of the Zero Lag Indicator suggest that an automatic crossover trading system is feasible using the Crossings of the EC and EMA lines as entry and exit signals for both long and short positions. However, a problem arises when the LeastError is relatively small. When the LeastError is small the EC and EMA have almost the same value, and therefore there are a considerable number of undesirable whipsaw trades that are made under this condition. The whipsaw trades can be reduced by setting an additional condition that the LeastError must be greater than some threshold value at the same time the EC and EMA lines cross. Figure 5 is the EasyLanguage code for the Zero Lag Trading System. In this code Thresh is an input parameter. To make the code more universal for all symbols the Threshold is compared to a percentage of the LeastError normalized to the closing price.

Inputs:

```
Length(20),
GainLimit(50),
Thresh(1);
```

Vars:

```
alpha(0),
Gain(0),
BestGain(0),
EC(0),
Error(0),
LeastError(0),
EMA(0);
```

```
alpha = 2 / (Length + 1);
```

```
EMA = alpha*Close + (1 - alpha)*EMA[1];
```

```
LeastError = 1000000;
```

```
For Value1 = -GainLimit to GainLimit Begin
```

```
    Gain = Value1 / 10;
```

```
    EC = alpha*(EMA + Gain*(Close - EC[1])) + (1 - alpha)*EC[1];
```

```
    Error = Close - EC;
```

```
    If AbsValue(Error) < LeastError Then Begin
```

```
        LeastError = AbsValue(Error);
```

```
        BestGain = Gain;
```

```
    End;
```

```
End;
```

```
EC = alpha*(EMA + BestGain*(Close - EC[1])) + (1 - alpha)*EC[1];
```

```
If EC Crosses Over EMA and 100*LeastError / Close > Thresh Then Buy Next Bar on Open;
```

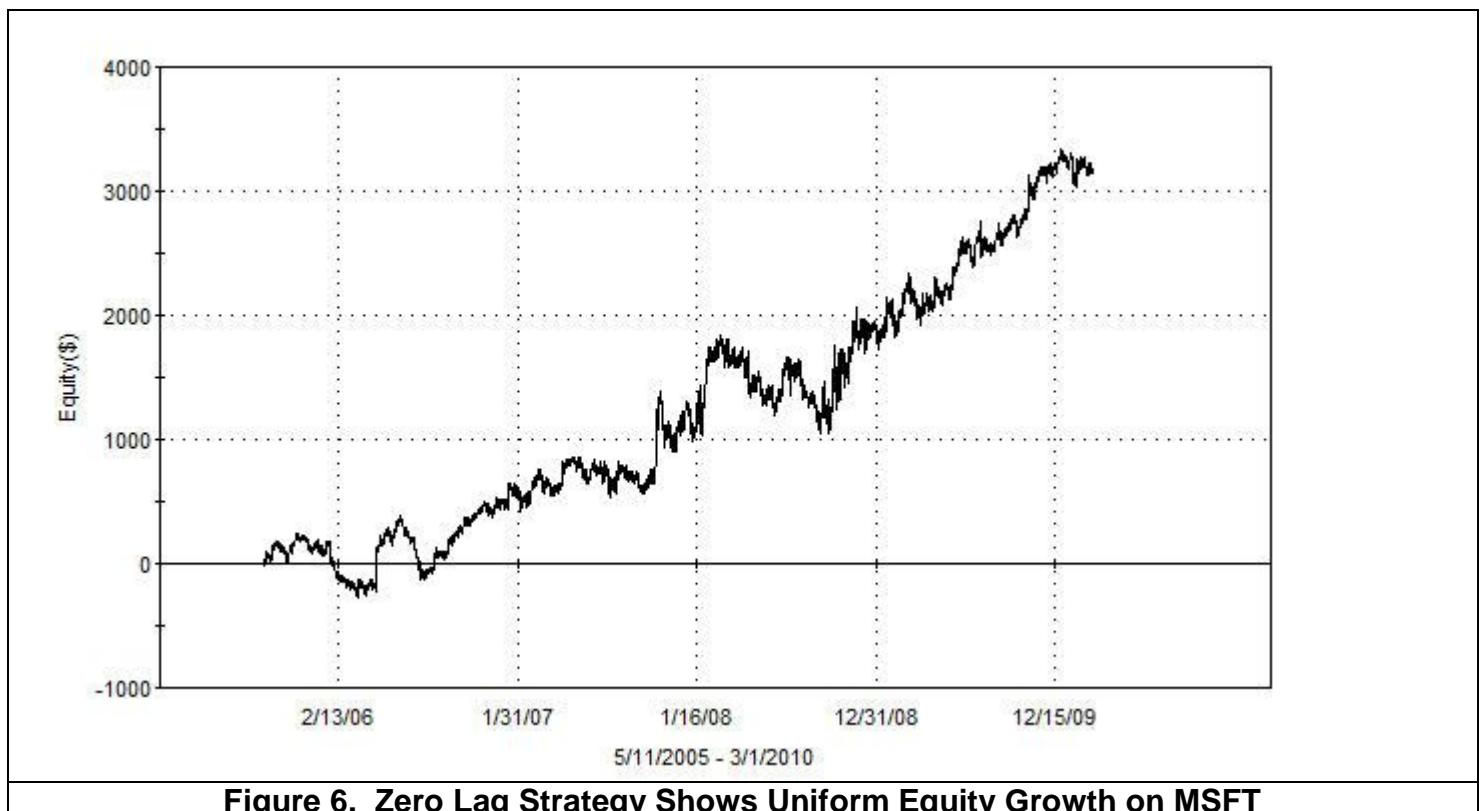
```
If EC Crosses Under EMA and 100*LeastError / Close > Thresh Then Sell Short Next Bar on Open;
```

**Figure 5. EasyLanguage Code for the Zero Lag Trading Strategy**

When the Zero Lag Strategy was applied to five years of 100 shares of MSFT with settings Length = 32, GainLimit = 22, and Thresh = 0.75 we obtained the following hypothetical results:

Net Profit	\$3,107
# Trades	20
Profit/Trade	\$155
% Profitable Trades	60%
Profit Factor	4.1

The equity curve over the five year period is shown In Figure 6. The relatively few trades over the five year observation period indicate the Zero Lag Trading Strategy is a Trend Follower. A selection of a shorter Period for the equivalent SMA length can turn it into a swing trader. If this is done in the case of MSFT, the percentage wins increases, the Profit Factor decreases, and the equity growth curve has less of a consistent growth than shown in Figure 6.



**Figure 6. Zero Lag Strategy Shows Uniform Equity Growth on MSFT**

Zero Lag is a new concept in adaptive technical analysis. This simple technique of applying a measured amount of error correction has resulted in both an effective technical indicator and automatic trading strategy that are free from the transients that usually accompany more sophisticated filters.