The CG Oscillator
By John Ehlers

INTRODUCTION

This article describes a new oscillator that is unique because it is both smoothed and has essentially zero lag. The smoothing enables clear identification of turning points and the zero lag aspect enables action to be taken early in the move. This oscillator is the serendipitous result of my research into adaptive filters. While the filters have not yet produced the result I seek, this oscillator has substantial advantages over conventional oscillators used in technical analysis. The "CG" in the name of the oscillator stands for the Center of Gravity of the prices over the window of observation.

The Center of Gravity (CG) of a physical object is its balance point. For example, if you balance a 12 inch ruler on your finger, the CG will be at its 6 inch point. If you change the weight distribution of the ruler by putting a paper clip on one end, then the balance point (e.g. the CG) shifts toward the paper clip. Moving from the physical world to the trading world, we can substitute the prices over our window of observation for the units of weight along the ruler. With this analogy, we see that the CG of the window moves to the right when prices increase sharply. Correspondingly, the CG of the window moves to the left when prices decrease.

COMPUTING THE CG

The idea of computing the Center of Gravity arose from observing how the lag of various FIR (Finite Impulse Response) filters vary according to the relative amplitude of the filter coefficients. A Simple Moving Average (SMA) is a FIR filter where all the filter coefficients have the same value (usually unity). As a result, the CG of the SMA is exactly in the center of the filter. A Weighted Moving Average (WMA) is an FIR filter where the most recent price is weighted by the length of the filter, the next most recent price is weighted by the length of the filter less one, and so on. The weighting terms are the filter coefficients. The filter coefficients of a WMA describe the outline of a triangle. It is well known that the CG of a triangle is located at one third the length of the base of the triangle. In other words, the CG of the WMA has shifted to the right relative to the DG of an SMA of equal length, resulting in less lag. In all cases of FIR filters the sum of the product of the coefficients and prices must be divided by the sum of the coefficients so that the scale of the original prices is retained.

The most general FIR filter is the Ehlers Filter[12] , which can be written as:

$$Ehlers\ Filt = \frac{\sum\limits_{i=0}^{N} c_i * \Pr ice_i}{\sum\limits_{i=0}^{N} c_i}$$

The coefficients of the Ehlers Filter can be almost any measure of variability. I have looked at momentum, signal to noise ratio, volatility, and even Stochastics and RSI values as filter coefficients. One of the most adaptive set of coefficients arose from video edge detection filters, and was the sum of the square of the differences of each price to each previous price. In any event, the result of using different filter coefficients is to make the filter adaptive by moving the CG of the coefficients.

While I was debugging the code of an adaptive FIR filter I noticed that the CG, itself, moved in exact opposition to the price swings. The CG moves to the right when prices go up and moves to the left when prices go down. Measured as the distance from the most recent price, the CG decreased when prices rose and increased when they fell. All I had to do was to invert the sign of the CG to get a smoothed oscillator that was both in phase with the price swings and had essentially zero lag.

The CG is computed in much the same way as we computed the Ehlers Filter. The position of the balance point is the summation of the product of position within the observation window times the price at that position divided by the summation of prices across the window. The mathematical expression for this calculation is:

$$CG = \frac{\sum\limits_{i=0}^{N} (x_i + 1) * \Pr ice_i}{\sum\limits_{i=0}^{N} \Pr ice_i}$$

In this expression we added one to the position count because we start our count with the most recent price at zero, and multiplying the most recent price by the position count would remove it from the computation. The EasyLanguage code to compute the CG Oscillator is given in Figure 1.
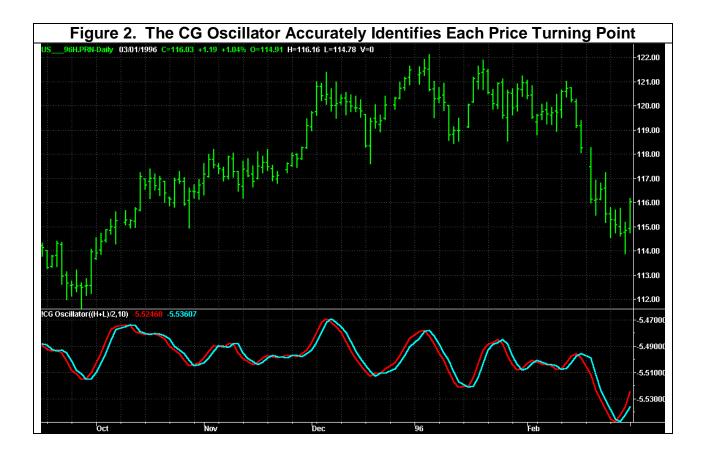
[1] John Ehlers, "Nonlinear Ehlers Filters", Stocks & Commodities, Apr 2001, page 25
[2] John Ehlers, "Rocket Science for Traders", John Wiley & Sons, New York, Chapter 10

Figure 1. Easy Language Code to Compute the CG Oscillator

```
Inputs:         Price((H+L)/2),
                Length(10);

Vars:           count(0),
                Num(0),
                Denom(0),
                CG(0);

Num = 0;
Denom = 0;
For count = 0 to Length - 1 begin
      Num = Num + (1 + count)*(Price[count]);
      Denom = Denom + (Price[count]);
End;
If Denom <> 0 then CG = -Num/Denom;

Plot1(CG, "CG");
Plot2(CG[1], "CG1");
```

In EasyLanguage, the notation Price[N] means the price "N" bars ago. Thus Price[0] is the price for the current bar. Counting for the location is backwards from the current bar. In the code the summation is accomplished by recursion, where the count is varied from the current bar to the length of the observation window. The numerator is the sum of the product of the bar position and the price, and the numerator is the sum of the prices. Then, the CG is just the negative ratio of the numerator to the denominator. Since the CG is smoothed, an effective crossover signal is produced simply by delaying the CG by one bar.

An example of the CG Oscillator is shown in Figure 2. In this case, I selected the length to be a 10 bar observation window. It is clear that every major price turning point is identified by the CG Oscillator and the crossovers formed by its trigger with zero lag. Since the CG Oscillator is filtered and smoothed, whipsaws of the crossovers are minimized.

The appearance of the CG Oscillator varies with the selection of the observation window length. Ideally, the selected length should be half the dominant cycle length because half the dominant cycle fully captures the entire cyclic move in one direction. If the length is too long, the CG Oscillator is desensitized. For example, if the window length is one full dominant cycle, half the data pulls the cg to the right and the other half the data pulls the cg to the left. As a result, the cg stays in the middle of the window and no motion of the CG oscillator is observed. On the other hand, if the window length is too short you are missing the benefits of smoothing. As a result of this case, the CG Oscillator contains higher frequency components and is a little too nervous for profitable trading.

Figure 2. The CG Oscillator Accurately Identifies Each Price Turning Point

John Ehlers is president of MESA Software and is a frequent contributor to Stocks & Commodities. He is a pioneer in introducing Maximum Entropy Spectral Analysis to technical traders through his MESA software. He is also a pioneer in introducing the Hilbert Transform with application to a number of unique indicators. He may be reached at (805) 969-6478, or via his websites www.mesasoftware.com or www.mesa-systems.com.