LIVE @ TAG

WORKBOOK

LIVE @ TAG

John Ehlers

Workshop

Full-Conference

Rocket Science for Traders

In this session John Ehlers will teach you to unleash the power of your computer by applying the science of modern digital signal processing to the art of trading. The result of this application is a series of indicators that operate together because they are adaptive to current market conditions. Computer code for all the indicators is provided.

Basic criteria are established for technical indicators, demonstrating their range and limitations. The concept of a Hilbert Transform is developed, and this transform is used to measure the dominant cycle and estimate the signal to noise ratio of the price data. The phase of the cyclic signal is extracted, and is used to derive a unique oscillator - The Sinewave Indicator. Additionally, the dominant cycle is removed from the data and the result is an Instantaneous Trendline. The way these new adaptive indicators work together is described.

Biography

John is an Electrical Engineer. He received his BSEE and MSEE from the University of Missouri and did his doctoral work at George Washington University, specializing in Fields & Waves and Information Theory. He has been a private trader since 1976, starting with fundamental analysis. With his engineering training he quickly gravitated to technical analysis of the market. He originally questioned what was magic about a 14 day RSI, or any other period. He concluded there was no unique answer and that variable market conditions must be measured for effective trading.

He discovered Maximum Entropy Spectrum Analysis (MESA) while attending an Information Theory seminar in 1978. He has been a pioneer in introducing the MESA concept into technical analysis for the measurement of short term cycles.

<< Next Page >>

Rocket Science for Trading

John Ehlers

MESA Software
P.O. Box 1801
Goleta, CA 93108

voice (800) 633-6372
FAX (805) 969-1358
john@mesasoftware.com

www.mesasoftware.com
www.mesa-systems.com

[<< Next Page >>]

RISK STATEMENT

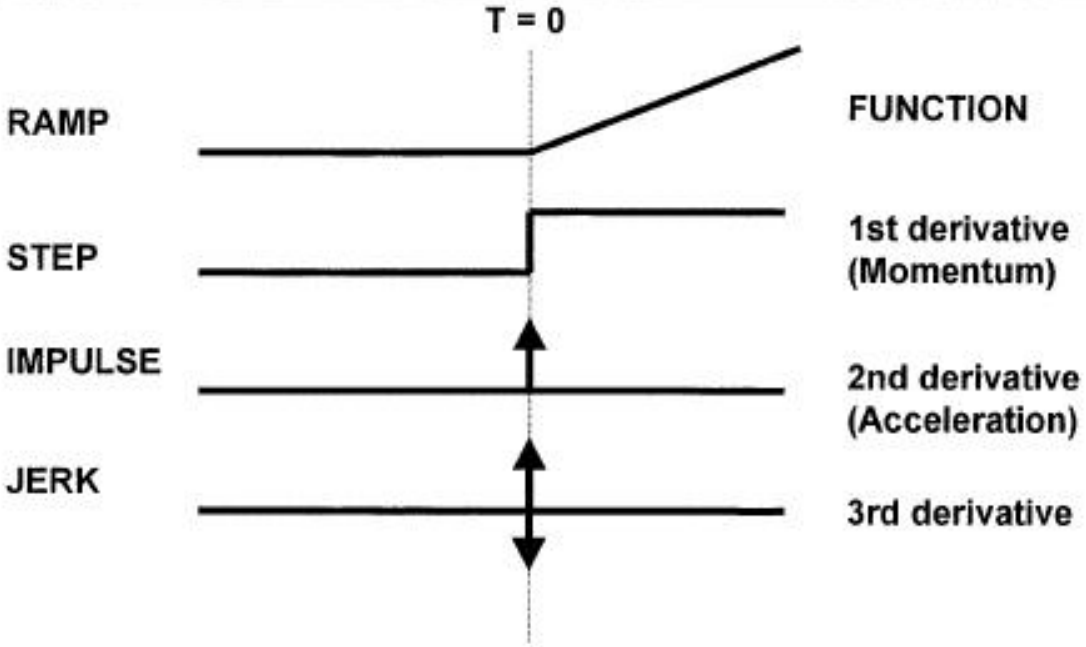
There is no guarantee that technical analysis will result in profits or that it will not result in losses. Past performance is not a guarantee of future results. All investments and trades carry risk and all trading decisions of an individual remain the responsibility of that individual. The use of any concepts presented does not constitute trading advice in any way.

AGENDA

- The character of Momentum Functions
- The character of Moving Averages
- Hilbert Transforms
- Measuring Cycle Periods
- Signal to Noise Indicator
- Signal Phase
- Sinewave Indicator
- Instantaneous Trendline
- The Awesome Oscillator
- Market Mode Detector

[<< Next Page >>]

Momentum Functions

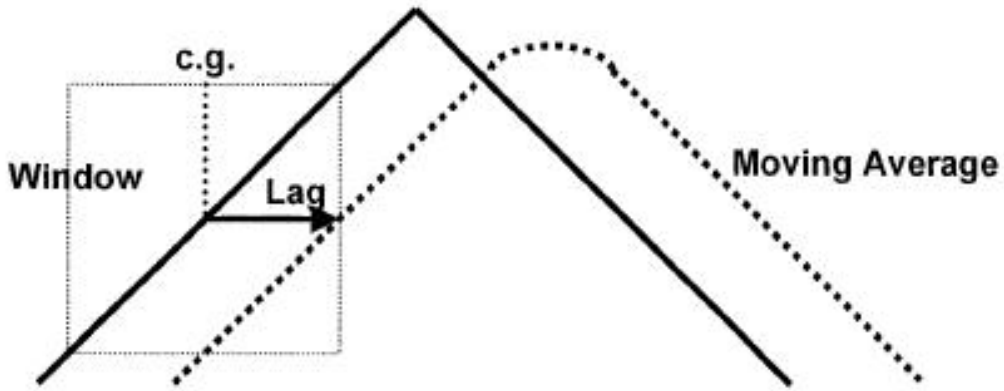


CONCLUSIONS:

1. Momentum can NEVER lead the function
2. Momentum is always more disjoint (noisy)

<< Next Page >>

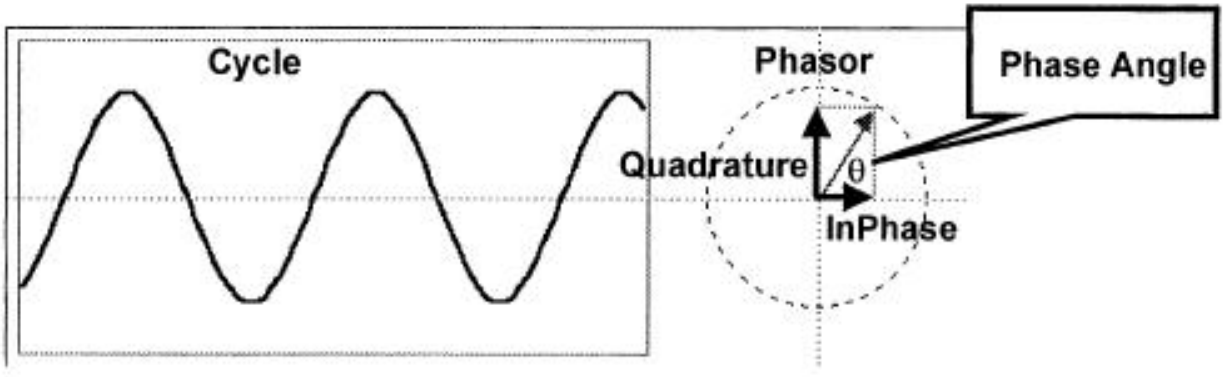
Moving Averages



CONCLUSIONS:

1. **Moving Averages smooth the function**
2. **Moving Averages Lag by the center of gravity of the observation window**
3. **Using Moving Averages is always a tradeoff between smoothing and lag**

A Phasor Describes a Cycle



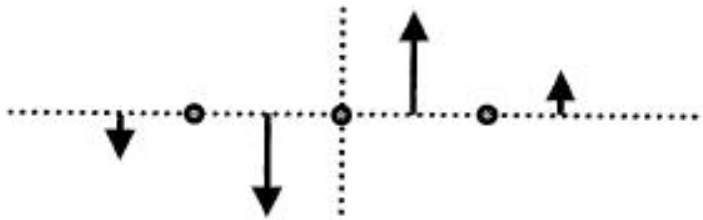
- **Cycle Amplitude (Pythagorean Theorem)**

  $$\text{Amplitude}^2 = (\text{InPhase})^2 + (\text{Quadrature})^2$$

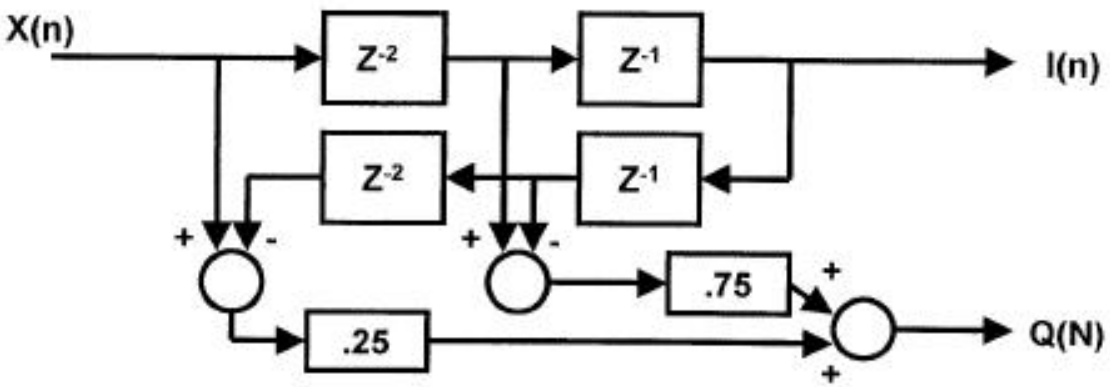
- **Phase Angle = ArcTan(Quadrature / InPhase)**
- **Cycle Period when Σ Phase Angles = 360^0**
- **So, the trick is to find the InPhase and Quadrature Components**

Hilbert TransformsYield I & Q

- Frequency response is exactly zero in the lower half of the unit circle
- $$h(n) = \frac{2Sin^2(\frac{\pi n}{2})}{\pi n}$$



- Implemented as a folded FIR Filter
  - Few terms are used to minimize lag



[<< Next Page >>]

Hilbert Transforms (cont)

- InPhase Component must have zero at zero frequency
  - Detrend Inphase component over the filter length
- EasyLanguage Code is:

```
Value1 = (4*Price + 3*Price[1] + 2*Price[2] + Price[3]) / 10;   {Smooth with one bar lag}
Value2 = (.25*Value1 + .75*Value1[2] - .75*Value1[4] - .25*Value1[5]);   {Detrend with linear phase}
Q1 = (.25*Value2 + .75*Value2[2] - .75*Value2[4] - .25*Value2[5]);   {Quadrature Component}
I1 = Value2[3];   {InPhase Component}
```


<< Next Page >>

Compensating Hilbert Transforms

- Differencing (momentum) is analogous to a calculus derivative
  - d(sin(ωt))/dt = ω∗Cos(ωt)
- Think about it
  - 1 bar delta of a 2 bar cycle is 2 (between +1 and -1)
  - maximum 1 bar delta of a 50 bar cycle is about 2/25 = .08
- Hilbert Detrender and Quadrature must be amplitude compensated
  - Use the period of the last bar since period changes slowly

```
Value1 = (4*Price + 3*Price[1] + 2*Price[2] + Price[3]) / 10;
Value2 = (.25*Value1 + .75*Value1[2] - .75*Value1[4] - .25*Value1[5])*(.05*Period[1] + .33);
Q1 = (.25*Value2 + .75*Value2[2] - .75*Value2[4] - .25*Value2[5])*(.05*Period[1] + .33);
I1 = Value2[3];
```

Measuring Cycle Period

- Use delay line discriminator for best performance in low signal to noise conditions
  - $Z(n) = X + jY = \rho e^{j(nt + \varphi)}$
  - $Z(n)^{*}Z^{*}(n-1) = \rho e^{j(\omega n + \varphi)} \rho e^{-j(\omega(n-1) + \varphi)} = \rho^{2} e^{j\omega}$
- Frequency is obtained by taking the arctangent of the argument of the data multiplied by the complex conjugate of the data delayed by one bar
- It is assumed there is only one dominant cycle present at a time

Cycle Period Code

CORE CODE

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

Vars:       I1(0), Q1(0), I2(0), Q2(0), X1(0), X2(0), Y1(0), Y2(0), Re(0), Im(0), ReAve(0), ImAve(0), Period(0);

If CurrentBar > 5 then begin

        {Compute InPhase and Quadrature components}
        Value1 = (3*Price + 2*Price[1] + Price[2])/6;
        Value2 = (.25*Value1 + .75*Value1[2] - .75*Value1[4] - .25*Value1[5])*(.05*Period[1] + .33);
        Q1 = (.25*Value2 + .75*Value2[2] - .75*Value2[4] - .25*Value2[5])*(.05*Period[1]+.33);
        I1 = Value2[3];
        {Smooth}
        Q2 = (4*Q1 + 3*Q1[1] + 2*Q1[2] + Q1[3])/10;
        I2 = (4*I1 + 3*I1[1] + 2*I1[2] + I1[3])/10;
        {Discriminator}
        X1 = I2*I2[1];
        X2 = I2*Q2[1];
        Y1 = Q2*Q2[1];
        Y2 = Q2*I2[1];
        Re = X1 + Y1;
        Im = X2 - Y2;
```

Cycle Period Code (cont)

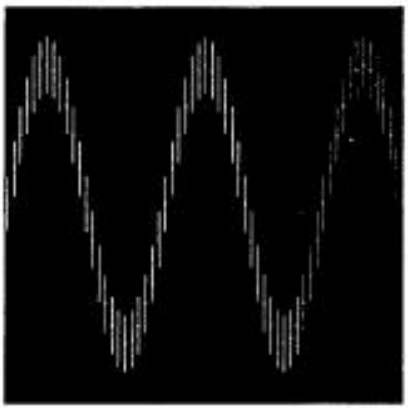
```
{Smooth Complex Product}
ReAve = (13*Re + 12*Re[1] + 11*Re[2] + 10*Re[3] + 9*Re[4] + 8*Re[5] + 7*Re[6] + 6*Re[7]
        + 5*Re[8] + 4*Re[9] + 3*Re[10] + 2*Re[11] + Re[12])/91;
ImAve = (13*Im + 12*Im[1] + 11*Im[2] + 10*Im[3] + 9*Im[4] + 8*Im[5] + 7*Im[6] + 6*Im[7]
        + 5*Im[8] + 4*Im[9] + 3*Im[10] + 2*Im[11] + Im[12])/91;
{Compute Dominant Cycle Period}
If ImAve <> 0 and ReAve <> 0 then Period = 360/ArcTangent(ImAve/ReAve);
{Limit bar-to-bar period variations to +/-50%}
If Period > 1.5*Period[1] then Period = 1.5*Period[1];
If Period < .67*Period[1] then Period = .67*Period[1];
{Bound absolute measurements}
If Period < 6 then Period = 6;
If Period > 50 then Period = 50;
{Smooth measured period}
Period = .2*Period + .8*Period[1];

end;
```

TO CORE CODE ADD:

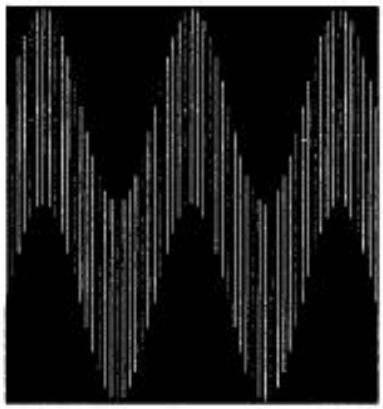
```
Plot1(Period, "Period");
```

Signal-to-Noise Indicator

- Noise is the uncertainty of a price sample
  - Can be pictured as the daily range on a chart
- Trading should only be done when signal is large relative to the noise
- 0 dB S/N occurs when the lowest low of a high signal point is equal to the highest high of a low signal point



High S/N



0 dB S/N

Signal-to-Noise Indicator Code

To CORE CODE ADD:

```
vars:       Signal(0), Noise(0);

{Signal Amplitude Squared comes directly from the discriminator}
Signal = SquareRoot(ReAve*ReAve + ImAve*ImAve);
{Noise is a 20 bar EMA of ranges squared}
Noise = .1*(H - L)*(H - L) + .9*Noise[1];
{5.4 term is corrected 6 dB value due to definition of signal amplitude}
If (Noise <> 0 and Signal <> 0) then SNR = .33*(10*Log(Signal/Noise)/Log(10) + 5.4) + .67*SNR[1];

Plot1(SNR, "SNR");
Plot2(6, "Ref");
```

Measuring Phase

- **Phase is measured at the period of the dominant cycle**

TO CORE CODE ADD

```
vars:       RealPart(0), ImagPart(0), DCPhase(0);

Value5 = IntPortion(Period);
RealPart = 0;
ImagPart = 0;
If Value5 > 6 then begin
            For count = 0 To Value5  - 1 begin
                        RealPart = RealPart + Sine(360 * count / Value5) * (Price[count]);
                        ImagPart = ImagPart + Cosine(360 * count / Value5) * (Price[count]);
            end;
end;
If AbsValue(ImagPart) > 0.001 then DCPhase = Arctangent(RealPart / ImagPart);
If AbsValue(ImagPart) <= 0.001 then DCPhase = 90 * Sign(RealPart);
DCPhase = DCPhase + 90;
If ImagPart < 0 then DCPhase = DCPhase + 180;
If DCPhase > 270 then DCPhase = DCPhase - 360;

Plot1(DCPhase, "Phase");
```

Using Phase - The Sinewave Indicator

- Code is exactly the same as the Phase Measurement except the Sine of the Phase Angle is Plotted
- Sine of Phase Angle plus 45 degrees is also plotted

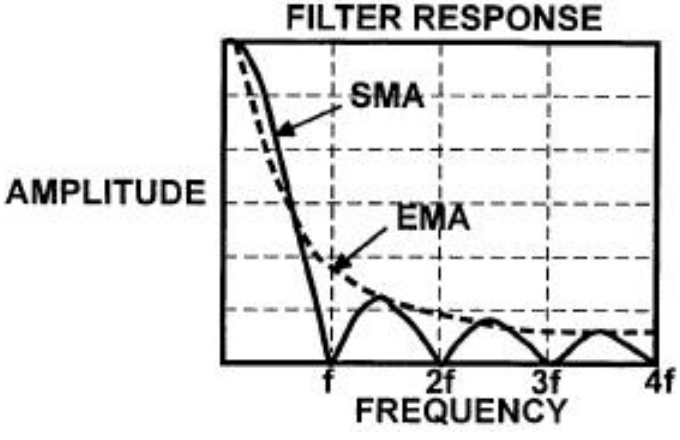
```
Plot1(Sine(DCPhase), "Sine");
Plot2(Sine(DCPhase + 45), "LeadSine");
```

Sinewave Indicator Advantages

- Line crossings give advance warning of cyclic turning points
- Advancing phase does not increase noise
- Indicator can be “tweaked” using theoretical waveforms
- No false whipsaws when the market is in a trend mode

Instantaneous Trendline

- Trend Mode exists when the phasor ceases to rotate - no rate change of phase
- Objective is to remove any dominant cycle component
  - Use a Simple Moving Average over the Dominant Cycle Period



<< Next Page >>

Instantaneous Trendline Code

- Trendline Lags Price by half the Dominant Cycle
- Trend is in force if Smooth fails to cross the Instantaneous Trendline within a half cycle period

ADD TO CORE CODE

```
vars:       Trend(0), Smooth(0);

Value5 = IntPortion(Period);
Trend = 0;
for count = 0 to Value5 - 1 begin
            Trend = Trend + Price[count];
end;
If Value5 <> 0 then Trend = Trend/Value5;
Trend = .25*Trend + .75*Trend[1];
Smooth = (7*Price + 6*Price[1] + 5*Price[2] + 4*Price[3] + 3*Price[4] + 2*Price[5] + Price[6])/28;

Plot1(Trend, "ITrend");
Plot2(Smooth, "Smooth");
```

The Awesome Oscillator

- Derived from RADAR Moving Target Indicator (MTI) circuits
- Handy to correlate with Sinewave Indicator
- Reference July 2000 Stocks & Commodities

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

Vars:       Detrend(0),
            Smooth(0);

Value1 = (4*Price + 3*Price[1] + 2*Price[3] + Price[4])/10;
Value2 = Value1 + .088*Value2[6];
Value3 = Value2 - Value2[6] + 1.2*Value3[6] - .7*Value3[12];
Detrend = Value3[12] - 2*Value3[6] + Value3;
Smooth = .13785*(2*Detrend - Detrend[1]) +.0007*(2*Detrend[1] - Detrend[2])
            + .13785*(2*Detrend[2] - Detrend[3]) + 1.2103*Smooth[1] - .4687*Smooth[2];

Plot1(Detrend,"Detrend");
Plot2(Smooth,"Smooth");
```

Market Mode Detector

- **Compile as a Paintbar**

TO CODE FOR PHASE ADD

```
vars:       trend(0), DaysInTrend(0);

{Rule 1:  Sinewave Indicator Crossing initializes half cycle crossing timing}
If Sine(DCPhase) Crosses Over Sine(DCPhase + 45) or
            Sine(DCPhase) Crosses Under Sine(DCPhase + 45) Then  begin
                        DaysInTrend = 0;
                        trend = 0;
end;
DaysInTrend = DaysInTrend + 1;
{Rule2:  Phase rate of change must be within 50% of dominant cycle phase rate of change}
If Value5 <> 0 and (DCPhase - DCPhase[1] > .67*360/Value5
            and DCPhase - DCPhase[1] < 1.5*360/Value5) then trend = 0;
If DaysInTrend > .5*Period then trend = 1;
If trend = 0 then begin
            Plot1(high, "high");
            Plot2(low, "low");
end;
```

And In Conclusion . . .

I know you believe you understood

what you think I said,

but I am not sure you realize

that what you heard is not what I meant

Thank you for taking this Live@TAG workshop

Speaker Info:

John Ehlers
Voice: 800.633.6372
Fax: 805 969.1358
www.mesasoftware.com

NEW workshops and speakers added on a regular basis.

Thank you for using

Live@TAG

Special *FREE* Offers from INO.com

Create your own ***FREE*** Personal Portfolio on INO.com.
http://quotes.ino.com/portfolio/

INO.com has many other ***FREE*** bonuses for you.
http://free.ino.com

Sign up for INO.com's ***FREE*** daily e-mail reports.
http://www.ino.com/email

Toll Free: 800.538.7424 • Overseas: 410.867.2100 • e-mail: live@ino.com