

```
{ Corona Chart Cycle Period  
written by John F. Ehlers    copyright (c) 2008
```

The spectral content of the data are measured in a bank of contiguous filters as described in "Measuring Cycle Periods" in the March 2008 issue of Stocks & Commodities Magazine. The filter having the strongest output is selected as the current dominant cycle period. The cycle period is measured as the number of bars contained in one full cycle period.

```
}
```

Inputs:

```
Price((H+L)/2),  
LineR(255),  
LineG(255),  
LineB(0),  
FuzzR(255),  
FuzzG(0),  
FuzzB(0),  
ShowDC(True);
```

Vars:

```
delta(0.1),  
gamma(0),  
alpha(0),  
beta(0),  
N(0),  
Period(0),  
MaxAmpl(0),  
Num(0),  
Denom(0),  
DC(0),  
DomCyc(0),  
Color1(0),  
Color2(0),  
Color3(0),  
alpha1(0),  
HP(0),  
SmoothHP(0);
```

Arrays:

```
I[60](0),  
OldI[60](0),  
OlderI[60](0),  
Q[60](0),  
OldQ[60](0),  
OlderQ[60](0),  
Real[60](0),  
OldReal[60](0),  
OlderReal[60](0),
```

```

    Imag[60](0),
    OldImag[60](0),
    OlderImag[60](0),
    Ampl[60](0),
    OldAmpl[60](0),
    DB[60](0),
    OldDB[60](0);

```

```

alpha1 = (1 - Sine (360 / 30)) / Cosine(360 / 30);
HP = .5*(1 + alpha1)*(Price - Price[1]) + alpha1*HP[1];
SmoothHP = (HP + 2*HP[1] + 3*HP[2] + 3*HP[3] + 2*HP[4] + HP[5]) / 12;
IF CurrentBar < 7 Then SmoothHP = Price - Price[1];
IF CurrentBar = 1 THEN SmoothHP = 0;

```

```

delta = -.015*CurrentBar + .5;
If delta < .1 then delta = .1;

```

```

If CurrentBar > 12 Then Begin
    For N = 12 to 60 Begin
        beta = Cosine(720 / N);
        gamma = 1 / Cosine(1440*delta / N);
        alpha = gamma - SquareRoot(gamma*gamma - 1);
        Q[N] = (.5*N / 6.28318)*(SmoothHP - SmoothHP[1]);
        I[N] = SmoothHP;
        Real[N] = .5*(1 - alpha)*(I[N] - OlderI[N]) + beta*(1 + alpha)*OldReal[N] -
alpha*OlderReal[N];
        Imag[N] = .5*(1 - alpha)*(Q[N] - OlderQ[N]) + beta*(1 + alpha)*OldImag[N] -
alpha*OlderImag[N];
        Ampl[N] = (Real[N]*Real[N] + Imag[N]*Imag[N]);
    End;
End;

```

```

For N = 12 to 60 Begin
    OlderI[N] = OldI[N];
    OldI[N] = I[N];
    OlderQ[N] = OldQ[N];
    OldQ[N] = Q[N];
    OlderReal[N] = OldReal[N];
    OldReal[N] = Real[N];
    OlderImag[N] = OldImag[N];
    OldImag[N] = Imag[N];
    OldAmpl[N] = Ampl[N];
    OldDB[N] = DB[N];
End;

```

```

MaxAmpl = Ampl[12];
For N = 12 to 60 Begin
    If Ampl[N] > MaxAmpl then MaxAmpl = Ampl[N];
End;

```

```

For N = 12 to 60 Begin
  //IF MaxAmpl <> 0 AND (Ampl[N] / MaxAmpl) > 0 THEN DB[N] = -60*Log(Ampl[N] /
MaxAmpl) / Log(10);
  IF MaxAmpl <> 0 AND (Ampl[N] / MaxAmpl) > 0 THEN DB[N] = -10*Log(.01 / (1 - .
99*Ampl[N] / MaxAmpl)) / Log(10);
    DB[N] = .33*DB[N] + .67*OldDB[N];
    If DB[N] > 20 then DB[N] = 20;
End;
Num = 0;
Denom = 0;
For N = 12 to 60 Begin
  If DB[N] <= 6 Then Begin
    Num = Num + N*(20 - DB[N]);
    Denom = Denom + (20 - DB[N]);
  End;
  If Denom <> 0 Then DC = .5*Num / Denom;
End;
DomCyc = Median(DC, 5);
If ShowDC = True Then Plot1(DomCyc, "DC", RGB(LineR, LineG, LineB), 0, 2);

```

```

For N = 12 to 60 Begin
  IF DB[N] <= 10 THEN Begin
    Color1 = LineR + DB[N]*(FuzzR - LineR) / 10;
    Color2 = LineG + DB[N]*(FuzzG - LineG) / 10;
    Color3 = LineB + DB[N]*(FuzzB - LineB) / 10;
  END;
  IF DB[N] > 10 THEN Begin
    Color1 = FuzzR*(2 - DB[N] / 10);
    Color2 = FuzzG*(2 - DB[N] / 10);
    Color3 = FuzzB*(2 - DB[N] / 10);
  END;
  If N = 12 Then Plot12(N/2, "S12", RGB(Color1, Color2, Color3),0,5);
  If N = 13 Then Plot13(N/2, "S13", RGB(Color1, Color2, Color3),0,5);
  If N = 14 Then Plot14(N/2, "S14", RGB(Color1, Color2, Color3),0,5);
  If N = 15 Then Plot15(N/2, "S15", RGB(Color1, Color2, Color3),0,5);
  If N = 16 Then Plot16(N/2, "S16", RGB(Color1, Color2, Color3),0,5);
  If N = 17 Then Plot17(N/2, "S17", RGB(Color1, Color2, Color3),0,5);
  If N = 18 Then Plot18(N/2, "S18", RGB(Color1, Color2, Color3),0,5);
  If N = 19 Then Plot19(N/2, "S19", RGB(Color1, Color2, Color3),0,5);
  If N = 20 Then Plot20(N/2, "S20", RGB(Color1, Color2, Color3),0,5);
  If N = 21 Then Plot21(N/2, "S21", RGB(Color1, Color2, Color3),0,5);
  If N = 22 Then Plot22(N/2, "S22", RGB(Color1, Color2, Color3),0,5);
  If N = 23 Then Plot23(N/2, "S23", RGB(Color1, Color2, Color3),0,5);
  If N = 24 Then Plot24(N/2, "S24", RGB(Color1, Color2, Color3),0,5);
  If N = 25 Then Plot25(N/2, "S25", RGB(Color1, Color2, Color3),0,5);
  If N = 26 Then Plot26(N/2, "S26", RGB(Color1, Color2, Color3),0,5);
  If N = 27 Then Plot27(N/2, "S27", RGB(Color1, Color2, Color3),0,5);
  If N = 28 Then Plot28(N/2, "S28", RGB(Color1, Color2, Color3),0,5);

```

```
If N = 29 Then Plot29(N/2, "S29", RGB(Color1, Color2, Color3),0,5);
If N = 30 Then Plot30(N/2, "S30", RGB(Color1, Color2, Color3),0,5);
If N = 31 Then Plot31(N/2, "S31", RGB(Color1, Color2, Color3),0,5);
If N = 32 Then Plot32(N/2, "S32", RGB(Color1, Color2, Color3),0,5);
If N = 33 Then Plot33(N/2, "S33", RGB(Color1, Color2, Color3),0,5);
If N = 34 Then Plot34(N/2, "S34", RGB(Color1, Color2, Color3),0,5);
If N = 35 Then Plot35(N/2, "S35", RGB(Color1, Color2, Color3),0,5);
If N = 36 Then Plot36(N/2, "S36", RGB(Color1, Color2, Color3),0,5);
If N = 37 Then Plot37(N/2, "S37", RGB(Color1, Color2, Color3),0,5);
If N = 38 Then Plot38(N/2, "S38", RGB(Color1, Color2, Color3),0,5);
If N = 39 Then Plot39(N/2, "S39", RGB(Color1, Color2, Color3),0,5);
If N = 40 Then Plot40(N/2, "S40", RGB(Color1, Color2, Color3),0,5);
If N = 41 Then Plot41(N/2, "S41", RGB(Color1, Color2, Color3),0,5);
If N = 42 Then Plot42(N/2, "S42", RGB(Color1, Color2, Color3),0,5);
If N = 43 Then Plot43(N/2, "S43", RGB(Color1, Color2, Color3),0,5);
If N = 44 Then Plot44(N/2, "S44", RGB(Color1, Color2, Color3),0,5);
If N = 45 Then Plot45(N/2, "S45", RGB(Color1, Color2, Color3),0,5);
If N = 46 Then Plot46(N/2, "S46", RGB(Color1, Color2, Color3),0,5);
If N = 47 Then Plot47(N/2, "S47", RGB(Color1, Color2, Color3),0,5);
If N = 48 Then Plot48(N/2, "S48", RGB(Color1, Color2, Color3),0,5);
If N = 49 Then Plot49(N/2, "S49", RGB(Color1, Color2, Color3),0,5);
If N = 50 Then Plot50(N/2, "S50", RGB(Color1, Color2, Color3),0,5);
If N = 51 Then Plot51(N/2, "S51", RGB(Color1, Color2, Color3),0,5);
If N = 52 Then Plot52(N/2, "S52", RGB(Color1, Color2, Color3),0,5);
If N = 53 Then Plot53(N/2, "S53", RGB(Color1, Color2, Color3),0,5);
If N = 54 Then Plot54(N/2, "S54", RGB(Color1, Color2, Color3),0,5);
If N = 55 Then Plot55(N/2, "S55", RGB(Color1, Color2, Color3),0,5);
If N = 56 Then Plot56(N/2, "S56", RGB(Color1, Color2, Color3),0,5);
If N = 57 Then Plot57(N/2, "S57", RGB(Color1, Color2, Color3),0,5);
If N = 58 Then Plot58(N/2, "S58", RGB(Color1, Color2, Color3),0,5);
If N = 59 Then Plot59(N/2, "S59", RGB(Color1, Color2, Color3),0,5);
If N = 60 Then Plot60(N/2, "S50", RGB(Color1, Color2, Color3),0,5);
```

End;

```
{ Corona Chart Signal to Noise Ratio
  written by John F. Ehlers    copyright (c) 2008
```

The amplitude of the dominant cycle is normalized to the amplitude of the noise. Noise is defined as the average daily trading range. The Signal to Noise Ratio is measured in decibels (dB). Unless the cycle amplitude is at least 3 dB greater than the noise, the use of cycle-based entries and oscillators should be avoided because the uncertainty of getting a good entry and exit point during day can negate any potential profit realized from the cyclic swing. A Signal to Noise ratio in excess of 6 dB signals a strong cyclic component relative to the noise.

```
}
```

Inputs:

```
Price((H+L)/2),
LineR(220),
LineG(255),
LineB(255),
FuzzR(0),
FuzzG(190),
FuzzB(190);
```

Vars:

```
delta(0.1),
gamma(0),
alpha(0),
beta(0),
N(0),
Period(0),
MaxAmpl(0),
Num(0),
Denom(0),
DC(0),
DomCyc(0),
Color1(0),
Color2(0),
Color3(0),
alpha1(0),
HP(0),
SmoothHP(0),
Avg(0),
Signal(0),
Noise(0),
SNR(0),
Width(0);
```

Arrays:

```
I[60](0),
OldI[60](0),
```

```

OlderI[60](0),
Q[60](0),
OldQ[60](0),
OlderQ[60](0),
Real[60](0),
OldReal[60](0),
OlderReal[60](0),
Imag[60](0),
OldImag[60](0),
OlderImag[60](0),
Ampl[60](0),
OldAmpl[60](0),
DB[60](0),
OldDB[60](0),
Raster[50](0),
OldRaster[50](0);

```

```

alpha1 = (1 - Sine (360 / 30)) / Cosine(360 / 30);
HP = .5*(1 + alpha1)*(Price - Price[1]) + alpha1*HP[1];
SmoothHP = (HP + 2*HP[1] + 3*HP[2] + 3*HP[3] + 2*HP[4] + HP[5]) / 12;
IF CurrentBar < 7 Then SmoothHP = Price - Price[1];
IF CurrentBar = 1 THEN SmoothHP = 0;

```

```

delta = -.015*CurrentBar + .5;
If delta < .1 then delta = .1;

```

```

If CurrentBar > 12 Then Begin
    For N = 12 to 60 Begin
        beta = Cosine(720 / N);
        gamma = 1 / Cosine(1440*delta / N);
        alpha = gamma - SquareRoot(gamma*gamma - 1);
        Q[N] = (.5*N / 6.28318)*(SmoothHP - SmoothHP[1]);
        I[N] = SmoothHP;
        Real[N] = .5*(1 - alpha)*(I[N] - OlderI[N]) + beta*(1 + alpha)*OldReal[N] -
alpha*OlderReal[N];
        Imag[N] = .5*(1 - alpha)*(Q[N] - OlderQ[N]) + beta*(1 + alpha)*OldImag[N] -
alpha*OlderImag[N];
        Ampl[N] = (Real[N]*Real[N] + Imag[N]*Imag[N]);
    End;
End;
For N = 12 to 60 Begin
    OlderI[N] = OldI[N];
    OldI[N] = I[N];
    OlderQ[N] = OldQ[N];
    OldQ[N] = Q[N];
    OlderReal[N] = OldReal[N];
    OldReal[N] = Real[N];
    OlderImag[N] = OldImag[N];
    OldImag[N] = Imag[N];

```

```

    OldAmpl[N] = Ampl[N];
    OldDB[N] = DB[N];
End;

For N = 1 to 50 Begin
    OldRaster[N] = Raster[N];
End;

MaxAmpl = Ampl[12];
For N = 12 to 60 Begin
    If Ampl[N] > MaxAmpl then MaxAmpl = Ampl[N];
End;

For N = 12 to 60 Begin
    If MaxAmpl <> 0 AND (Ampl[N] / MaxAmpl) > 0 Then DB[N] = -10*Log(.01 / (1 - .99*Ampl[N] /
MaxAmpl)) / Log(10);
    DB[N] = .33*DB[N] + .67*OldDB[N];
    If DB[N] > 20 then DB[N] = 20;
End;
Num = 0;
Denom = 0;
For N = 12 to 60 Begin
    If DB[N] <= 6 Then Begin
        Num = Num + N*(20 - DB[N]);
        Denom = Denom + (20 - DB[N]);
    End;
    If Denom <> 0 Then DC = .5*Num / Denom;
End;
DomCyc = Median(DC, 5);
If DomCyc < 6 Then DomCyc = 6;

Avg = .1*Price + .9*Avg[1];
If Avg <> 0 and MaxAmpl > 0 Then Signal = .2*SquareRoot(MaxAmpl) + .9*Signal[1];
If Avg <> 0 and CurrentBar > 5 Then Noise = .1*Median((H-L), 5) + .9*Noise[1];
If Signal <> 0 and Noise <> 0 Then SNR = 20*Log(Signal / Noise) / Log(10) + 3.5;
IF SNR < 1 Then SNR = 0;
If SNR > 11 Then SNR = 10;
SNR = .1*SNR;

Width = -.4*SNR + .2;
If SNR > .5 then Width = 0;

For N = 1 to 50 Begin
    Raster[N] = 20;
    If N < Round(50*SNR,0) Then Raster[N] = .5*(Power((20*SNR - .4*N)/ Width, .8) +
OldRaster[N]);
    If N > Round(50*SNR, 0) and (.4*N - 20*SNR) / Width > 1 Then Raster[N] = .
5*(Power((-20*SNR + .4*N)/ Width, .8) + OldRaster[N]) ;
    If N = Round(50*SNR,0) Then Raster[N] = 0 + .5*OldRaster[N];

```

```

If Raster[N] < 0 Then Raster[N] = 0;
If Raster[N] > 20 Then Raster[N] = 20;
If SNR > .5 then Raster[N] = 20;

```

```
End;
```

```
Plot1(10*SNR+1, "S51", RGB(LineR, LineG, LineB),0,2);
```

```
For N = 1 to 50 Begin
```

```
IF Raster[N] <= 10 THEN Begin
```

```
Color1 = LineR + Raster[N]*(FuzzR - LineR) / 10;
```

```
Color2 = LineG + Raster[N]*(FuzzG - LineG) / 10;
```

```
Color3 = LineB + Raster[N]*(FuzzB - LineB) / 10;
```

```
END;
```

```
IF Raster[N] > 10 THEN Begin
```

```
Color1 = FuzzR*(2 - Raster[N] / 10);
```

```
Color2 = FuzzG*(2 - Raster[N] / 10);
```

```
Color3 = FuzzB*(2 - Raster[N] / 10);
```

```
END;
```

```
//If N = 1 Then Plot1(N, "S1", RGB(Color1, Color2, Color3),Color3,5);
```

```
If N = 2 Then Plot2(.2*N+1, "S2", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 3 Then Plot3(.2*N+1, "S3", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 4 Then Plot4(.2*N+1, "S4", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 5 Then Plot5(.2*N+1, "S5", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 6 Then Plot6(.2*N+1, "S6", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 7 Then Plot7(.2*N+1, "S7", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 8 Then Plot8(.2*N+1, "S8", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 9 Then Plot9(.2*N+1, "S9", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 10 Then Plot10(.2*N+1, "S10", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 11 Then Plot11(.2*N+1, "S11", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 12 Then Plot12(.2*N+1, "S12", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 13 Then Plot13(.2*N+1, "S13", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 14 Then Plot14(.2*N+1, "S14", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 15 Then Plot15(.2*N+1, "S15", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 16 Then Plot16(.2*N+1, "S16", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 17 Then Plot17(.2*N+1, "S17", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 18 Then Plot18(.2*N+1, "S18", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 19 Then Plot19(.2*N+1, "S19", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 20 Then Plot20(.2*N+1, "S20", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 21 Then Plot21(.2*N+1, "S21", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 22 Then Plot22(.2*N+1, "S22", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 23 Then Plot23(.2*N+1, "S23", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 24 Then Plot24(.2*N+1, "S24", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 25 Then Plot25(.2*N+1, "S25", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 26 Then Plot26(.2*N+1, "S26", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 27 Then Plot27(.2*N+1, "S27", RGB(Color1, Color2, Color3),0,5);
```

```
If N = 28 Then Plot28(.2*N+1, "S28", RGB(Color1, Color2, Color3),0,5);
```



```
If N = 29 Then Plot29(.2*N+1, "S29", RGB(Color1, Color2, Color3),0,5);
If N = 30 Then Plot30(.2*N+1, "S30", RGB(Color1, Color2, Color3),0,5);
If N = 31 Then Plot31(.2*N+1, "S31", RGB(Color1, Color2, Color3),0,5);
If N = 32 Then Plot32(.2*N+1, "S32", RGB(Color1, Color2, Color3),0,5);
If N = 33 Then Plot33(.2*N+1, "S33", RGB(Color1, Color2, Color3),0,5);
If N = 34 Then Plot34(.2*N+1, "S34", RGB(Color1, Color2, Color3),0,5);
If N = 35 Then Plot35(.2*N+1, "S35", RGB(Color1, Color2, Color3),0,5);
If N = 36 Then Plot36(.2*N+1, "S36", RGB(Color1, Color2, Color3),0,5);
If N = 37 Then Plot37(.2*N+1, "S37", RGB(Color1, Color2, Color3),0,5);
If N = 38 Then Plot38(.2*N+1, "S38", RGB(Color1, Color2, Color3),0,5);
If N = 39 Then Plot39(.2*N+1, "S39", RGB(Color1, Color2, Color3),0,5);
If N = 40 Then Plot40(.2*N+1, "S40", RGB(Color1, Color2, Color3),0,5);
If N = 41 Then Plot41(.2*N+1, "S41", RGB(Color1, Color2, Color3),0,5);
If N = 42 Then Plot42(.2*N+1, "S42", RGB(Color1, Color2, Color3),0,5);
If N = 43 Then Plot43(.2*N+1, "S43", RGB(Color1, Color2, Color3),0,5);
If N = 44 Then Plot44(.2*N+1, "S44", RGB(Color1, Color2, Color3),0,5);
If N = 45 Then Plot45(.2*N+1, "S45", RGB(Color1, Color2, Color3),0,5);
If N = 46 Then Plot46(.2*N+1, "S46", RGB(Color1, Color2, Color3),0,5);
If N = 47 Then Plot47(.2*N+1, "S47", RGB(Color1, Color2, Color3),0,5);
If N = 48 Then Plot48(.2*N+1, "S48", RGB(Color1, Color2, Color3),0,5);
If N = 49 Then Plot49(.2*N+1, "S49", RGB(Color1, Color2, Color3),0,5);
If N = 50 Then Plot50(.2*N+1, "S50", RGB(Color1, Color2, Color3),0,5);
```

End;

```
{ Corona Chart Swing Position  
  written by John F. Ehlers    copyright (c) 2008
```

The swing position indicator shows the phasing of the data within the dominant cycle. A value of -5 means the cycle is at its valley. A value of +5 means the cycle is at its peak. In a pure cycle the Swing Position will trace out the shape of a sine wave.

```
}
```

Inputs:

```
Price((H+L)/2),  
LineR(180),  
LineG(255),  
LineB(210),  
FuzzR(0),  
FuzzG(172),  
FuzzB(64);
```

Vars:

```
delta(0.1),  
gamma(0),  
alpha(0),  
beta(0),  
N(0),  
Period(0),  
MaxAmpl(0),  
Num(0),  
Denom(0),  
DC(0),  
DomCyc(0),  
Color1(0),  
Color2(0),  
Color3(0),  
alpha1(0),  
HP(0),  
SmoothHP(0),  
gamma2(0),  
alpha2(0),  
beta2(0),  
delta2(.1),  
BP2(0),  
Q2(0),  
Lead60(0),  
HL(0),  
LL(0),  
count(0),  
Psn(0),  
Width(0);
```

Arrays:

```
I[60](0),
OldI[60](0),
OlderI[60](0),
Q[60](0),
OldQ[60](0),
OlderQ[60](0),
Real[60](0),
OldReal[60](0),
OlderReal[60](0),
Imag[60](0),
OldImag[60](0),
OlderImag[60](0),
Ampl[60](0),
OldAmpl[60](0),
DB[60](0),
OldDB[60](0),
Raster[50](0),
OldRaster[50](0);
```

```
alpha1 = (1 - Sine (360 / 30)) / Cosine(360 / 30);
HP = .5*(1 + alpha1)*(Price - Price[1]) + alpha1*HP[1];
SmoothHP = (HP + 2*HP[1] + 3*HP[2] + 3*HP[3] + 2*HP[4] + HP[5]) / 12;
IF CurrentBar < 7 Then SmoothHP = Price - Price[1];
IF CurrentBar = 1 THEN SmoothHP = 0;
```

```
delta = -.015*CurrentBar + .5;
If delta < .1 then delta = .1;
```

```
If CurrentBar > 12 Then Begin
```

```
For N = 12 to 60 Begin
```

```
beta = Cosine(720 / N);
```

```
gamma = 1 / Cosine(1440*delta / N);
```

```
alpha = gamma - SquareRoot(gamma*gamma - 1);
```

```
Q[N] = (.5*N / 6.28318)*(SmoothHP - SmoothHP[1]);
```

```
I[N] = SmoothHP;
```

```
Real[N] = .5*(1 - alpha)*(I[N] - OlderI[N]) + beta*(1 + alpha)*OldReal[N] -
alpha*OlderReal[N];
```

```
Imag[N] = .5*(1 - alpha)*(Q[N] - OlderQ[N]) + beta*(1 + alpha)*OldImag[N] -
alpha*OlderImag[N];
```

```
Ampl[N] = (Real[N]*Real[N] + Imag[N]*Imag[N]);
```

```
End;
```

```
End;
```

```
For N = 12 to 60 Begin
```

```
OlderI[N] = OldI[N];
```

```
OldI[N] = I[N];
```

```
OlderQ[N] = OldQ[N];
```

```
OldQ[N] = Q[N];
```

```
OlderReal[N] = OldReal[N];
```

```

    OldReal[N] = Real[N];
    OlderImag[N] = OldImag[N];
    OldImag[N] = Imag[N];
    OldAmpl[N] = Ampl[N];
    OldDB[N] = DB[N];
End;

For N = 1 to 50 Begin
    OldRaster[N] = Raster[N];
End;

MaxAmpl = Ampl[12];
For N = 12 to 60 Begin
    If Ampl[N] > MaxAmpl then MaxAmpl = Ampl[N];
End;

For N = 12 to 60 Begin
    If MaxAmpl <> 0 AND (Ampl[N] / MaxAmpl) > 0 Then DB[N] = -10*Log(.01 / (1 - .99*Ampl[N] /
MaxAmpl)) / Log(10);
    DB[N] = .33*DB[N] + .67*OldDB[N];
    If DB[N] > 20 then DB[N] = 20;
End;
Num = 0;
Denom = 0;
For N = 12 to 60 Begin
    If DB[N] <= 6 Then Begin
        Num = Num + N*(20 - DB[N]);
        Denom = Denom + (20 - DB[N]);
    End;
    If Denom <> 0 Then DC = .5*Num / Denom;
End;
DomCyc = Median(DC, 5);
If DomCyc < 6 Then DomCyc = 6;

beta2 = Cosine(360 / DomCyc);
gamma2 = 1 / Cosine(720*delta2 / DomCyc);
alpha2 = gamma2 - SquareRoot(gamma2*gamma2 - 1);
BP2 = .5*(1 - alpha2)*(Price - Price[2]) + beta2*(1 + alpha2)*BP2[1] - alpha2*BP2[2];
Q2 = (DomCyc / 6.28318)*(BP2 - BP2[1]);
Lead60 = .5*BP2 + .866*Q2;

HL = Lead60;
LL = Lead60;
For count = 0 to 50 Begin
    If Lead60[count] > HL then HL = Lead60[count];
    If Lead60[count] < LL then LL = Lead60[count];
End;
Psn = (Lead60 - LL) / (HL - LL);

```

```

HL = Psn;
LL = Psn;
For count = 0 to 20 Begin
    If Psn[count] > HL then HL = Psn[count];
    If Psn[count] < LL then LL = Psn[count];
End;
If HL - LL > .85 Then Width = .01 Else Width = .15*(HL - LL);

For N = 1 to 50 Begin
    Raster[N] = 20;
    If N < Round(50*Psn,0) Then Raster[N] = .5*(Power((20*Psn - .4*N)/ Width, .95) + .
5*OldRaster[N]);
    If N > Round(50*Psn,0) Then Raster[N] = .5*(Power((-20*Psn + .4*N)/ Width, .95) + .
5*OldRaster[N]);
    If N = Round(50*Psn,0) Then Raster[N] = .5*OldRaster[N];
    //If Date = 20080401 Then Print(Date);
    If Raster[N] < 0 Then Raster[N] = 0;
    If Raster[N] > 20 Then Raster[N] = 20;
    If HL - LL > .8 then Raster[N] = 20;
    OldRaster[N] = Raster[N];
End;

Plot1(10*Psn-5, "S51", RGB(LineR, LineG, LineB),0,2);

For N = 1 to 50 Begin

    IF Raster[N] <= 10 THEN Begin
        Color1 = LineR + Raster[N]*(FuzzR - LineR) / 10;
        Color2 = LineG + Raster[N]*(FuzzG - LineG) / 10;
        Color3 = LineB + Raster[N]*(FuzzB - LineB) / 10;
    END;
    IF Raster[N] > 10 THEN Begin
        Color1 = FuzzR*(2 - Raster[N] / 10);
        Color2 = FuzzG*(2 - Raster[N] / 10);
        Color3 = FuzzB*(2 - Raster[N] / 10);

    END;

    //If N = 1 Then Plot1(N, "S1", RGB(Color1, Color2, Color3),Color3,5);
    If N = 2 Then Plot2(.2*N-5, "S2", RGB(Color1, Color2, Color3),0,5);
    If N = 3 Then Plot3(.2*N-5, "S3", RGB(Color1, Color2, Color3),0,5);
    If N = 4 Then Plot4(.2*N-5, "S4", RGB(Color1, Color2, Color3),0,5);
    If N = 5 Then Plot5(.2*N-5, "S5", RGB(Color1, Color2, Color3),0,5);
    If N = 6 Then Plot6(.2*N-5, "S6", RGB(Color1, Color2, Color3),0,5);
    If N = 7 Then Plot7(.2*N-5, "S7", RGB(Color1, Color2, Color3),0,5);
    If N = 8 Then Plot8(.2*N-5, "S8", RGB(Color1, Color2, Color3),0,5);
    If N = 9 Then Plot9(.2*N-5, "S9", RGB(Color1, Color2, Color3),0,5);
    If N = 10 Then Plot10(.2*N-5, "S10", RGB(Color1, Color2, Color3),0,5);
    If N = 11 Then Plot11(.2*N-5, "S11", RGB(Color1, Color2, Color3),0,5);

```

```
If N = 12 Then Plot12(.2*N-5, "S12", RGB(Color1, Color2, Color3),0,5);
If N = 13 Then Plot13(.2*N-5, "S13", RGB(Color1, Color2, Color3),0,5);
If N = 14 Then Plot14(.2*N-5, "S14", RGB(Color1, Color2, Color3),0,5);
If N = 15 Then Plot15(.2*N-5, "S15", RGB(Color1, Color2, Color3),0,5);
If N = 16 Then Plot16(.2*N-5, "S16", RGB(Color1, Color2, Color3),0,5);
If N = 17 Then Plot17(.2*N-5, "S17", RGB(Color1, Color2, Color3),0,5);
If N = 18 Then Plot18(.2*N-5, "S18", RGB(Color1, Color2, Color3),0,5);
If N = 19 Then Plot19(.2*N-5, "S19", RGB(Color1, Color2, Color3),0,5);
If N = 20 Then Plot20(.2*N-5, "S20", RGB(Color1, Color2, Color3),0,5);
If N = 21 Then Plot21(.2*N-5, "S21", RGB(Color1, Color2, Color3),0,5);
If N = 22 Then Plot22(.2*N-5, "S22", RGB(Color1, Color2, Color3),0,5);
If N = 23 Then Plot23(.2*N-5, "S23", RGB(Color1, Color2, Color3),0,5);
If N = 24 Then Plot24(.2*N-5, "S24", RGB(Color1, Color2, Color3),0,5);
If N = 25 Then Plot25(.2*N-5, "S25", RGB(Color1, Color2, Color3),0,5);
If N = 26 Then Plot26(.2*N-5, "S26", RGB(Color1, Color2, Color3),0,5);
If N = 27 Then Plot27(.2*N-5, "S27", RGB(Color1, Color2, Color3),0,5);
If N = 28 Then Plot28(.2*N-5, "S28", RGB(Color1, Color2, Color3),0,5);
If N = 29 Then Plot29(.2*N-5, "S29", RGB(Color1, Color2, Color3),0,5);
If N = 30 Then Plot30(.2*N-5, "S30", RGB(Color1, Color2, Color3),0,5);
If N = 31 Then Plot31(.2*N-5, "S31", RGB(Color1, Color2, Color3),0,5);
If N = 32 Then Plot32(.2*N-5, "S32", RGB(Color1, Color2, Color3),0,5);
If N = 33 Then Plot33(.2*N-5, "S33", RGB(Color1, Color2, Color3),0,5);
If N = 34 Then Plot34(.2*N-5, "S34", RGB(Color1, Color2, Color3),0,5);
If N = 35 Then Plot35(.2*N-5, "S35", RGB(Color1, Color2, Color3),0,5);
If N = 36 Then Plot36(.2*N-5, "S36", RGB(Color1, Color2, Color3),0,5);
If N = 37 Then Plot37(.2*N-5, "S37", RGB(Color1, Color2, Color3),0,5);
If N = 38 Then Plot38(.2*N-5, "S38", RGB(Color1, Color2, Color3),0,5);
If N = 39 Then Plot39(.2*N-5, "S39", RGB(Color1, Color2, Color3),0,5);
If N = 40 Then Plot40(.2*N-5, "S40", RGB(Color1, Color2, Color3),0,5);
If N = 41 Then Plot41(.2*N-5, "S41", RGB(Color1, Color2, Color3),0,5);
If N = 42 Then Plot42(.2*N-5, "S42", RGB(Color1, Color2, Color3),0,5);
If N = 43 Then Plot43(.2*N-5, "S43", RGB(Color1, Color2, Color3),0,5);
If N = 44 Then Plot44(.2*N-5, "S44", RGB(Color1, Color2, Color3),0,5);
If N = 45 Then Plot45(.2*N-5, "S45", RGB(Color1, Color2, Color3),0,5);
If N = 46 Then Plot46(.2*N-5, "S46", RGB(Color1, Color2, Color3),0,5);
If N = 47 Then Plot47(.2*N-5, "S47", RGB(Color1, Color2, Color3),0,5);
If N = 48 Then Plot48(.2*N-5, "S48", RGB(Color1, Color2, Color3),0,5);
If N = 49 Then Plot49(.2*N-5, "S49", RGB(Color1, Color2, Color3),0,5);
If N = 50 Then Plot50(.2*N-5, "S50", RGB(Color1, Color2, Color3),0,5);
```

End;

{ Corona Chart Trend Vigor  
written by John F. Ehlers      copyright(c) 2008

Once the dominant cycle period is known, the instantaneous trend is just the momentum across the full cycle period. This measurement is invariant with the position within the cycle. The trend slope is the same whether measured from cycle peak to cycle peak or cycle valley to cycle valley. The trend slope is normalized to the amplitude of the dominant cycle. A value of +2 means the trend slope is twice the dominant cycle amplitude, and therefore cautions you not to trade against the upslope. Similarly, a value of -2 means the trend slope is down and therefore cautions you not to buy against the downslope.

}

Inputs:

Price((H+L)/2),  
LineR(64),  
LineG(128),  
LineB(255),  
FuzzR(0),  
FuzzG(0),  
FuzzB(255);

Vars:

delta(0.1),  
gamma(0),  
alpha(0),  
beta(0),  
N(0),  
Period(0),  
MaxAmpl(0),  
Num(0),  
Denom(0),  
DC(0),  
DomCyc(0),  
Color1(0),  
Color2(0),  
Color3(0),  
alpha1(0),  
HP(0),  
SmoothHP(0),  
delta1(.1),  
gamma1(0),  
alpha2(0),  
beta1(0),  
IP(0),  
Q1(0),  
Ampl2(0),  
Trend(0),

```

Ratio(0),
TV(0),
Width(0);

```

Arrays:

```

I[60](0),
OldI[60](0),
OlderI[60](0),
Q[60](0),
OldQ[60](0),
OlderQ[60](0),
Real[60](0),
OldReal[60](0),
OlderReal[60](0),
Imag[60](0),
OldImag[60](0),
OlderImag[60](0),
Ampl[60](0),
OldAmpl[60](0),
DB[60](0),
OldDB[60](0),
Raster[50](0),
OldRaster[50](0);

```

```

alpha1 = (1 - Sine (360 / 30)) / Cosine(360 / 30);
HP = .5*(1 + alpha1)*(Price - Price[1]) + alpha1*HP[1];
SmoothHP = (HP + 2*HP[1] + 3*HP[2] + 3*HP[3] + 2*HP[4] + HP[5]) / 12;
IF CurrentBar < 7 Then SmoothHP = Price - Price[1];
IF CurrentBar = 1 THEN SmoothHP = 0;

```

```

delta = -.015*CurrentBar + .5;
If delta < .1 then delta = .1;

```

If CurrentBar > 12 Then Begin

For N = 12 to 60 Begin

```

    beta = Cosine(720 / N);
    gamma = 1 / Cosine(1440*delta / N);
    alpha = gamma - SquareRoot(gamma*gamma - 1);
    Q[N] = (.5*N / 6.28318)*(SmoothHP - SmoothHP[1]);
    I[N] = SmoothHP;

```

```

    Real[N] = .5*(1 - alpha)*(I[N] - OlderI[N]) + beta*(1 + alpha)*OldReal[N] -
alpha*OlderReal[N];

```

```

    Imag[N] = .5*(1 - alpha)*(Q[N] - OlderQ[N]) + beta*(1 + alpha)*OldImag[N] -
alpha*OlderImag[N];

```

```

    Ampl[N] = (Real[N]*Real[N] + Imag[N]*Imag[N]);

```

End;

End;

For N = 12 to 60 Begin

```

    OlderI[N] = OldI[N];

```



```

    OldI[N] = I[N];
    OlderQ[N] = OldQ[N];
    OldQ[N] = Q[N];
    OlderReal[N] = OldReal[N];
    OldReal[N] = Real[N];
    OlderImag[N] = OldImag[N];
    OldImag[N] = Imag[N];
    OldAmpl[N] = Ampl[N];
    OldDB[N] = DB[N];
End;

For N = 1 to 50 Begin
    OldRaster[N] = Raster[N];
End;

MaxAmpl = Ampl[12];
For N = 12 to 60 Begin
    If Ampl[N] > MaxAmpl then MaxAmpl = Ampl[N];
End;

For N = 12 to 60 Begin
    If MaxAmpl <> 0 AND (Ampl[N] / MaxAmpl) > 0 Then DB[N] = -10*Log(.01 / (1 - .99*Ampl[N] /
MaxAmpl)) / Log(10);
    DB[N] = .33*DB[N] + .67*OldDB[N];
    If DB[N] > 20 then DB[N] = 20;
End;
Num = 0;
Denom = 0;
For N = 12 to 60 Begin
    If DB[N] <= 6 Then Begin
        Num = Num + N*(20 - DB[N]);
        Denom = Denom + (20 - DB[N]);
    End;
    If Denom <> 0 Then DC = .5*Num / Denom;
End;
DomCyc = Median(DC, 5);
If DomCyc < 6 Then DomCyc = 6;

//Filter Bandpass component
beta1 = Cosine(360 / DomCyc);
gamma1 = 1 / Cosine(720*delta1 / DomCyc);
alpha2 = gamma1 - SquareRoot(gamma1*gamma1 - 1);
IP = .5*(1 - alpha2)*(Price - Price[2]) + beta1*(1 + alpha2)*IP[1] - alpha2*IP[2];

//Quadrature component is derivative of InPhase component divided by omega
Q1 = (Domcyc / 6.28318)*(IP - IP[1]);

//Pythagorean theorem to establish cycle amplitude
Ampl2 = SquareRoot(IP*IP + Q1*Q1);

```

```

//Trend amplitude taken over the cycle period
Trend = Price - Price[DomCyc - 1];
If Trend <> 0 and Ampl2 <> 0 Then Ratio = .33*Trend / Ampl2 + .67*Ratio[1];
If Ratio > 10 then Ratio = 10;
If Ratio < -10 then Ratio = -10;
TV = .05*(Ratio + 10);

If TV < .3 or TV > .7 Then Width = .01;
If TV >= .3 and TV < .5 Then Width = TV - .3;
If TV > .5 and TV <= .7 Then Width = -TV + .7;

For N = 1 to 50 Begin
    Raster[N] = 20;
    If N < Round(50*TV,0) Then Raster[N] = .8*(Power((20*TV - .4*N)/ Width, .85) + .
2*OldRaster[N]);
    If N > Round(50*TV,0) Then Raster[N] = .8*(Power((-20*TV + .4*N)/ Width, .85) + .
2*OldRaster[N]);
    If N = Round(50*TV,0) Then Raster[N] = 0 + .5*OldRaster[N];
    If Raster[N] < 0 Then Raster[N] = 0;
    If Raster[N] > 20 Or TV < .3 Or TV > .7 Then Raster[N] = 20;
End;

Plot1(20*TV-10, "S51", RGB(LineR, LineG, LineB),0,2);

For N = 1 to 50 Begin

    IF Raster[N] <= 10 THEN Begin
        Color1 = LineR + Raster[N]*(FuzzR - LineR) / 10;
        Color2 = LineG + Raster[N]*(FuzzG - LineG) / 10;
        Color3 = LineB + Raster[N]*(FuzzB - LineB) / 10;
    END;
    IF Raster[N] > 10 THEN Begin
        Color1 = FuzzR*(2 - Raster[N] / 10);
        Color2 = FuzzG*(2 - Raster[N] / 10);
        Color3 = FuzzB*(2 - Raster[N] / 10);
    END;

    //If N = 1 Then Plot1(N, "S1", RGB(Color1, Color2, Color3),Color3,5);
    If N = 2 Then Plot2(.4*N-10, "S2", RGB(Color1, Color2, Color3),0,5);
    If N = 3 Then Plot3(.4*N-10, "S3", RGB(Color1, Color2, Color3),0,5);
    If N = 4 Then Plot4(.4*N-10, "S4", RGB(Color1, Color2, Color3),0,5);
    If N = 5 Then Plot5(.4*N-10, "S5", RGB(Color1, Color2, Color3),0,5);
    If N = 6 Then Plot6(.4*N-10, "S6", RGB(Color1, Color2, Color3),0,5);
    If N = 7 Then Plot7(.4*N-10, "S7", RGB(Color1, Color2, Color3),0,5);
    If N = 8 Then Plot8(.4*N-10, "S8", RGB(Color1, Color2, Color3),0,5);
    If N = 9 Then Plot9(.4*N-10, "S9", RGB(Color1, Color2, Color3),0,5);
    If N = 10 Then Plot10(.4*N-10, "S10", RGB(Color1, Color2, Color3),0,5);

```

```
If N = 11 Then Plot11(.4*N-10, "S11", RGB(Color1, Color2, Color3),0,5);
If N = 12 Then Plot12(.4*N-10, "S12", RGB(Color1, Color2, Color3),0,5);
If N = 13 Then Plot13(.4*N-10, "S13", RGB(Color1, Color2, Color3),0,5);
If N = 14 Then Plot14(.4*N-10, "S14", RGB(Color1, Color2, Color3),0,5);
If N = 15 Then Plot15(.4*N-10, "S15", RGB(Color1, Color2, Color3),0,5);
If N = 16 Then Plot16(.4*N-10, "S16", RGB(Color1, Color2, Color3),0,5);
If N = 17 Then Plot17(.4*N-10, "S17", RGB(Color1, Color2, Color3),0,5);
If N = 18 Then Plot18(.4*N-10, "S18", RGB(Color1, Color2, Color3),0,5);
If N = 19 Then Plot19(.4*N-10, "S19", RGB(Color1, Color2, Color3),0,5);
If N = 20 Then Plot20(.4*N-10, "S20", RGB(Color1, Color2, Color3),0,5);
If N = 21 Then Plot21(.4*N-10, "S21", RGB(Color1, Color2, Color3),0,5);
If N = 22 Then Plot22(.4*N-10, "S22", RGB(Color1, Color2, Color3),0,5);
If N = 23 Then Plot23(.4*N-10, "S23", RGB(Color1, Color2, Color3),0,5);
If N = 24 Then Plot24(.4*N-10, "S24", RGB(Color1, Color2, Color3),0,5);
If N = 25 Then Plot25(.4*N-10, "S25", RGB(Color1, Color2, Color3),0,5);
If N = 26 Then Plot26(.4*N-10, "S26", RGB(Color1, Color2, Color3),0,5);
If N = 27 Then Plot27(.4*N-10, "S27", RGB(Color1, Color2, Color3),0,5);
If N = 28 Then Plot28(.4*N-10, "S28", RGB(Color1, Color2, Color3),0,5);
If N = 29 Then Plot29(.4*N-10, "S29", RGB(Color1, Color2, Color3),0,5);
If N = 30 Then Plot30(.4*N-10, "S30", RGB(Color1, Color2, Color3),0,5);
If N = 31 Then Plot31(.4*N-10, "S31", RGB(Color1, Color2, Color3),0,5);
If N = 32 Then Plot32(.4*N-10, "S32", RGB(Color1, Color2, Color3),0,5);
If N = 33 Then Plot33(.4*N-10, "S33", RGB(Color1, Color2, Color3),0,5);
If N = 34 Then Plot34(.4*N-10, "S34", RGB(Color1, Color2, Color3),0,5);
If N = 35 Then Plot35(.4*N-10, "S35", RGB(Color1, Color2, Color3),0,5);
If N = 36 Then Plot36(.4*N-10, "S36", RGB(Color1, Color2, Color3),0,5);
If N = 37 Then Plot37(.4*N-10, "S37", RGB(Color1, Color2, Color3),0,5);
If N = 38 Then Plot38(.4*N-10, "S38", RGB(Color1, Color2, Color3),0,5);
If N = 39 Then Plot39(.4*N-10, "S39", RGB(Color1, Color2, Color3),0,5);
If N = 40 Then Plot40(.4*N-10, "S40", RGB(Color1, Color2, Color3),0,5);
If N = 41 Then Plot41(.4*N-10, "S41", RGB(Color1, Color2, Color3),0,5);
If N = 42 Then Plot42(.4*N-10, "S42", RGB(Color1, Color2, Color3),0,5);
If N = 43 Then Plot43(.4*N-10, "S43", RGB(Color1, Color2, Color3),0,5);
If N = 44 Then Plot44(.4*N-10, "S44", RGB(Color1, Color2, Color3),0,5);
If N = 45 Then Plot45(.4*N-10, "S45", RGB(Color1, Color2, Color3),0,5);
If N = 46 Then Plot46(.4*N-10, "S46", RGB(Color1, Color2, Color3),0,5);
If N = 47 Then Plot47(.4*N-10, "S47", RGB(Color1, Color2, Color3),0,5);
If N = 48 Then Plot48(.4*N-10, "S48", RGB(Color1, Color2, Color3),0,5);
If N = 49 Then Plot49(.4*N-10, "S49", RGB(Color1, Color2, Color3),0,5);
If N = 50 Then Plot50(.4*N-10, "S50", RGB(Color1, Color2, Color3),0,5);
```

End;