FOURIER TRANSFORM FOR TRADERS

By John Ehlers

It is intrinsically wrong to use a 14 bar RSI, a 9 bar Stochastic, a 5/25 Double Moving Average crossover, or any other fixed-length indicator when the market conditions are variable. It's kind of like driving on a curvy, foggy mountain road with your cruise control locked. That market conditions are continuously changing is not even a subject of debate. There have been a number of attempts to adapt to changing market conditions. Volatility-based nonlinear moving averages are just one example to adapt to market changes. Coming from an information theory background, my answer to the question of how to adapt to changing conditions is to first measure the dominant market cycle and then tune the various indicators to that cycle period, or a fraction of it. For example, an RSI theoretically performs best when the computation period is just one half of a cycle period. That is, when all the movement is in one direction and then reverses so all the movement is in the other direction over the period of one cycle – and you get a full amplitude swing from the RSI.

Make no mistake. Measuring market cycles is difficult. Not only the problem of simultaneously solving for frequency, amplitude, and phase to arrive at an accurate estimate, but also one must realized the measurement is being made in a low signal to noise environment. Further, one must be concerned with the responsiveness of the measuring technique to capture the cycle periods that are continuously changing without introducing transient artifacts into the measurement. There is a variety of spectrum estimation techniques available, ranging from the Fourier Periodogram to modern high resolution spectral analysis approaches.

I have long railed against the use of Fourier Transforms for use in estimating market cycles because of their lack of resolution. I'll show you what I mean. Figure 1 represents a typical spectrum measurement. The horizontal axis is the frequency (or its reciprocal – cycle period) scale. The vertical axis is the amplitude scale. The frequency having the highest amplitude identifies the measured cycle. If the width of the spectral line is narrow, just being a spike like the solid line, then the cycle is identified with high resolution. If we have a high resolution technique we could, in fact, identify two closely spaced cycle periods if they are present in the data. On the other hand, if we have a low resolution measurement technique, the width of the spectral line is very broad, two closely spaced cycles could be averaged together and you could not identify them, demonstrated by the dotted line. On the right hand vertical scale of Figure 1 you see a color bar. I convert the amplitude to colors ranging from white hot to ice cold, through red hot. Conversion to color enables me to view the spectrum directly below the price barchart as a heatmap. If the spectrum measurement is highly focused with high resolution, then the spectral display appears basically as a yellow line. If the spectral line is very wide, then it appears as a yellow blob in the heatmap display.

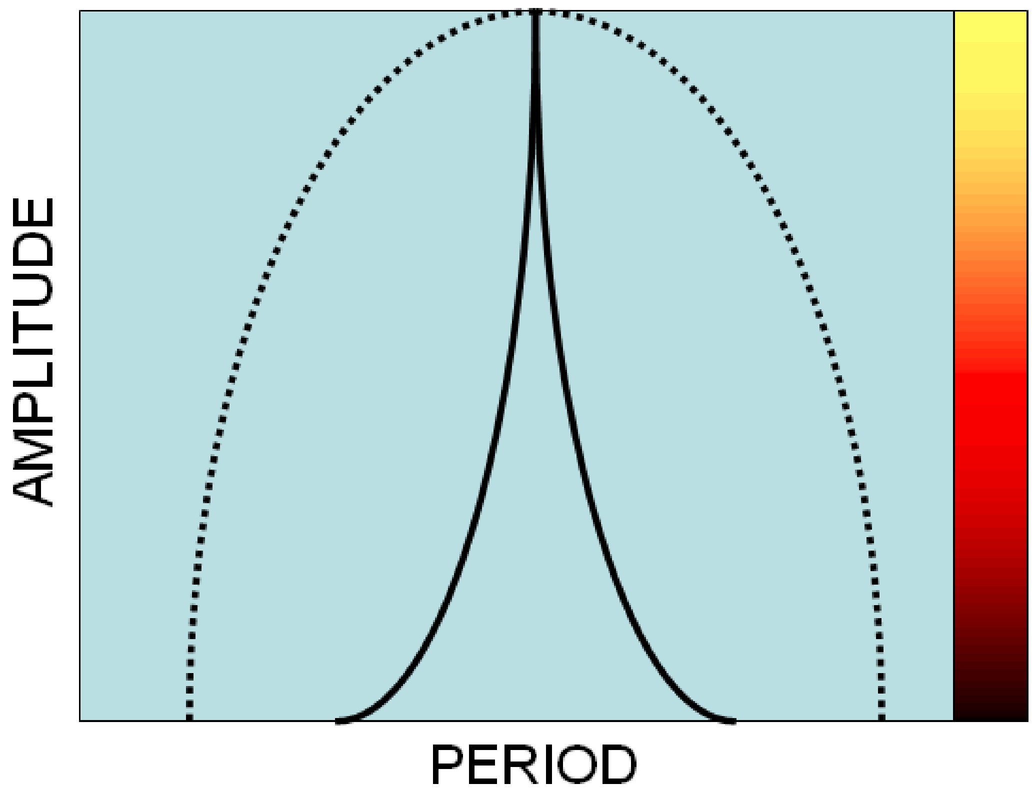

Figure 1. Spectrum Conversion

Figure 2 shows the spectrum measured by a Discrete Fourier Transform (DFT) below the barchart for IBM. The color in the heatmap indicates the cycle amplitude and the cycle period is the vertical scale, scaled from 8 to 50 bars at the right hand side of the chart. The heatmap is in time synchronism with the barchart. Now I think you see why I have advised against the use of Fourier Transforms. There may be some cyclic activity, but it is so blurry that it cannot be useful for trading. It is kind of like driving in a dense fog – maybe you can see something, but it would be a better idea to pull off the road.

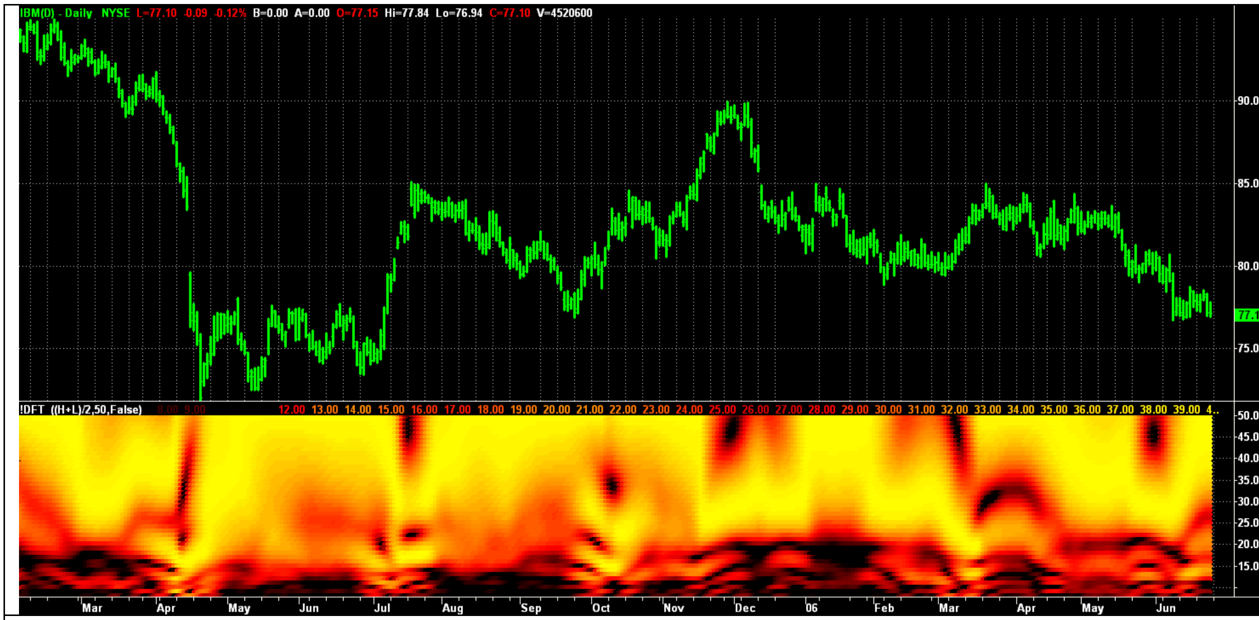

Figure 2. DFT of IBM Has Very Poor Spectral Resolution

More recently I came across a relatively obscure paper during the course of my research on high resolution spectral analysis techniques.[1] Kay and Demeure showed the spectrums of Figure 3 (I have taken artistic license and have redrawn them. That is, my drawings are not precisely accurate – but they convey the concept). These are the spectra produced by the more modern MUSIC (Multiple Signal Classification) algorithm and the spectrum produced by a Barlett (Fourier-type) algorithm. Kay and Demeure posed the question of which had the highest resolution. The two spectral lines are obvious in the MUSIC spectrum, but not in the Bartlett spectrum. One would conclude from this visual inspection that the Bartlett spectrum is vastly inferior with respect to resolution.

[1] Steven Kay and Cedric Demeure, "The High-Resolution Spectrum Estimator – a Subjective Entity", Proceedings IEEE, Vol 72, Dec 1984, pp1815-1816

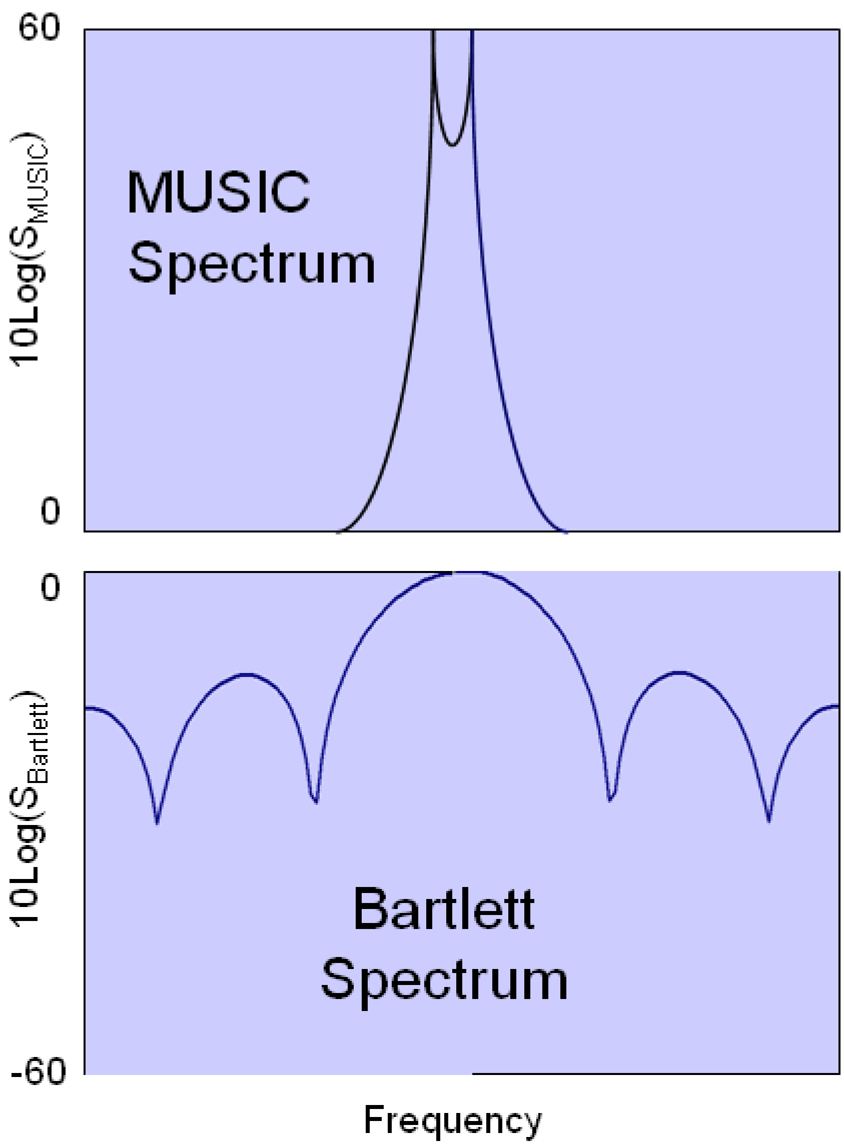

Figure 3. MUSIC and Bartlett Spectral Estimate Visual Comparison Implies that MUSIC Has a Greater Resolution

In the referent paper, Kay & Demeure showed that the two spectra are simply related by the transformation:

$$S_{MUSIC} = \frac{1}{1 - S_{Bartlett}}$$

Where $0 \leq S_{Bartlett} \leq 1$

So, when the Barlett spectrum is 1, the denominator of the transformation goes to zero and the spectral line of the MUSIC spectrum goes to infinity. Of course, infinity is never reached in real world measurements. Nonetheless, the point is that the two techniques have the same resolution regardless of how they look in a visual inspection when the transform is taken into account.

Armed with this transform, I applied it to the DFT spectrum estimator using a 20 decibel amplification (100:1) of the peak spectral line. The result is a spectrum that is now useful for identifying the variable dominant cycle in the market, as shown in Figure 4. In my opinion, when the cycle period exceeds 50 bars, the market is in a trend and the cycle measurement can be of little help. Therefore, I limit the display to be between 8 and 50 bars.

Figure 4. Transformed DFT Spectral Estimate of IBM More Clearly Identifies the Variable Dominant Cycle

The EasyLanguage code to apply the DFT, apply the transformation, and plot the spectrum is given in Figure 5. Preprocessing is important for spectral analysis to avoid having undesired frequency components introduce cross products in the multiply steps of the analysis. Therefore, I detrend the data by first passing it through a 40 bar highpass filter. Since long cycles are rejected by the highpass filter, the end effect errors of having noninteger cycles within the data window is relatively small. The highpass filter is only a 2 pole filter, and so components out to our 50 bar plotting limit are passed. I eliminate the 2 bar, 3 bar, and 4 bar cycle components by low pass filtering in a 6 element FIR filter. After the DFT portion I also show how to extract the dominant cycle using a center of gravity algorithm. My experience is that this center of gravity approach yields the smoothest and most reliable estimate of the dominant cycle period. Traders converting this code to other platforms probably will have difficulty displaying the spectrum. However, extraction of the dominant cycle does not depend on spectrum plotting. The entire spectrum is computed before the dominant cycle is extracted. Note the spectral estimate of Figure 4 confirms that there is only one strong cycle present in the data most of the time. Simultaneous cycles are present with a low probability. Therefore, the concept of using a dominant cycle to tune indicators is valid, or at least sufficiently valid to be used to your advantage to dynamically adjust your indicators.

```
{                       Discrete Fourier Transform
                        Copyright (c) 2006 John F. Ehlers
}

Inputs:
        Price((H+L)/2),
        Window(50),
        ShowDC(False);

Vars:
        alpha1(0),
        HP(0),
        CleanedData(0),
        Period(0),
        n(0),
        MaxPwr(0),
        Num(0),
        Denom(0),
        DominantCycle(0),
        Color1(0),
        Color2(0);

//Arrays are sized to have a maximum Period of 50 bars
Arrays:
        CosinePart[50](0),
        SinePart[50](0),
        Pwr[50](0),
        DB[50](0);

//Get a detrended version of the data by High Pass Filtering with a 40 Period cutoff
If CurrentBar <= 5 Then Begin
        HP = Price;
        CleanedData = Price;
End;
If CurrentBar > 5 Then Begin
   alpha1 = (1 - Sine(360/40))/Cosine(360/40);
   HP = .5*(1 + alpha1)*(Price - Price[1]) + alpha1*HP[1];
   CleanedData = (HP + 2*HP[1] + 3*HP[2] + 3*HP[3] + 2*HP[4] + HP[5])/12;
End;

//This is the DFT
```

```
For Period = 8 to 50 Begin
	CosinePart[Period] = 0;
	SinePart[Period] = 0;
   FOR n = 0 to Window - 1 Begin
		CosinePart[Period] = CosinePart[Period] + CleanedData[n]*Cosine(360*n/Period);
		SinePart[Period] = SinePart[Period] + CleanedData[n]*Sine(360*n/Period);
	End;
	Pwr[Period] = CosinePart[Period]*CosinePart[Period] + SinePart[Period]*SinePart[Period];
End;

//Find Maximum Power Level for Normalization
MaxPwr = Pwr[8];
For Period = 8 to 50 Begin
	If Pwr[Period] > MaxPwr Then MaxPwr = Pwr[Period];
End;

//Normalize Power Levels and Convert to Decibels
For Period = 8 to 50 Begin
	IF MaxPwr > 0 and Pwr[Period] > 0 Then DB[Period] = -10*LOG(.01 / (1 - .99*Pwr[Period] / MaxPwr))/Log(10);
	If DB[Period] > 20 then DB[Period] = 20;
End;

//Find Dominant Cycle using CG algorithm
Num = 0;
Denom = 0;
For Period = 8 to 50 Begin
	If DB[Period] < 3 Then Begin
		Num = Num + Period*(3 - DB[Period]);
		Denom = Denom + (3 - DB[Period]);
	End;
End;
If Denom <> 0 then DominantCycle = Num/Denom;
If ShowDC = True Then Plot1(DominantCycle, "S1", RGB(0, 0, 255),0,2);

//Plot the Spectrum as a Heatmap
For Period = 8 to 50 Begin
	//Convert Decibels to RGB Color for Display
	If DB[Period] > 10 Then Begin
		Color1 = 255*(2 - DB[Period]/10);
		Color2 = 0;
	End;
	If DB[Period] <= 10 Then Begin
		Color1 = 255;
```

```
            Color2 = 255*(1 - DB[Period]/10);
        End;
        If Period = 8 Then Plot8(8, "S8", RGB(Color1, Color2, 0),0,4);
        If Period = 9 Then Plot9(9, "S9", RGB(Color1, Color2, 0),0,4);
        If Period = 10 Then Plot10(10, "S10", RGB(Color1, Color2, 0),0,4);
        If Period = 11 Then Plot11(11, "S11", RGB(Color1, Color2, 0),0,4);
        If Period = 12 Then Plot12(12, "S12", RGB(Color1, Color2, 0),0,4);
        If Period = 13 Then Plot13(13, "S13", RGB(Color1, Color2, 0),0,4);
        If Period = 14 Then Plot14(14, "S14", RGB(Color1, Color2, 0),0,4);
        If Period = 15 Then Plot15(15, "S15", RGB(Color1, Color2, 0),0,4);
        If Period = 16 Then Plot16(16, "S16", RGB(Color1, Color2, 0),0,4);
        If Period = 17 Then Plot17(17, "S17", RGB(Color1, Color2, 0),0,4);
        If Period = 18 Then Plot18(18, "S18", RGB(Color1, Color2, 0),0,4);
        If Period = 19 Then Plot19(19, "S19", RGB(Color1, Color2, 0),0,4);
        If Period = 20 Then Plot20(20, "S20", RGB(Color1, Color2, 0),0,4);
        If Period = 21 Then Plot21(21, "S21", RGB(Color1, Color2, 0),0,4);
        If Period = 22 Then Plot22(22, "S22", RGB(Color1, Color2, 0),0,4);
        If Period = 23 Then Plot23(23, "S23", RGB(Color1, Color2, 0),0,4);
        If Period = 24 Then Plot24(24, "S24", RGB(Color1, Color2, 0),0,4);
        If Period = 25 Then Plot25(25, "S25", RGB(Color1, Color2, 0),0,4);
        If Period = 26 Then Plot26(26, "S26", RGB(Color1, Color2, 0),0,4);
        If Period = 27 Then Plot27(27, "S27", RGB(Color1, Color2, 0),0,4);
        If Period = 28 Then Plot28(28, "S28", RGB(Color1, Color2, 0),0,4);
        If Period = 29 Then Plot29(29, "S29", RGB(Color1, Color2, 0),0,4);
        If Period = 30 Then Plot30(30, "S30", RGB(Color1, Color2, 0),0,4);
        If Period = 31 Then Plot31(31, "S31", RGB(Color1, Color2, 0),0,4);
        If Period = 32 Then Plot32(32, "S32", RGB(Color1, Color2, 0),0,4);
        If Period = 33 Then Plot33(33, "S33", RGB(Color1, Color2, 0),0,4);
        If Period = 34 Then Plot34(34, "S34", RGB(Color1, Color2, 0),0,4);
        If Period = 35 Then Plot35(35, "S35", RGB(Color1, Color2, 0),0,4);
        If Period = 36 Then Plot36(36, "S36", RGB(Color1, Color2, 0),0,4);
        If Period = 37 Then Plot37(37, "S37", RGB(Color1, Color2, 0),0,4);
        If Period = 38 Then Plot38(38, "S38", RGB(Color1, Color2, 0),0,4);
        If Period = 39 Then Plot39(39, "S39", RGB(Color1, Color2, 0),0,4);
        If Period = 40 Then Plot40(40, "S40", RGB(Color1, Color2, 0),0,4);
        If Period = 41 Then Plot41(41, "S41", RGB(Color1, Color2, 0),0,4);
        If Period = 42 Then Plot42(42, "S42", RGB(Color1, Color2, 0),0,4);
        If Period = 43 Then Plot43(43, "S43", RGB(Color1, Color2, 0),0,4);
        If Period = 44 Then Plot44(44, "S44", RGB(Color1, Color2, 0),0,4);
        If Period = 45 Then Plot45(45, "S45", RGB(Color1, Color2, 0),0,4);
        If Period = 46 Then Plot46(46, "S46", RGB(Color1, Color2, 0),0,4);
        If Period = 47 Then Plot47(47, "S47", RGB(Color1, Color2, 0),0,4);
        If Period = 48 Then Plot48(48, "S48", RGB(Color1, Color2, 0),0,4);
        If Period = 49 Then Plot49(49, "S49", RGB(Color1, Color2, 0),0,4);
        If Period = 50 Then Plot50(50, "S50", RGB(Color1, Color2, 0),0,4);
End;
```

Figure 5. Transformed DFT EasyLanguage Code

Kay and Demeure conclude that means other than visual inspection should be used to assess the resolution of spectral estimators. One approach is to create known data that contains two closely spaced cycles and see if the spectral estimator can discern whether there are two cycles present or not. I therefore created a set of data containing a pure 20 bar cycle and a pure 24 bar cycle of equal amplitude. Figure 5 shows the measurement of the DFT spectral estimator. In a nutshell, it does not have sufficient resolution to identify that both these closely spaced cycles are present.

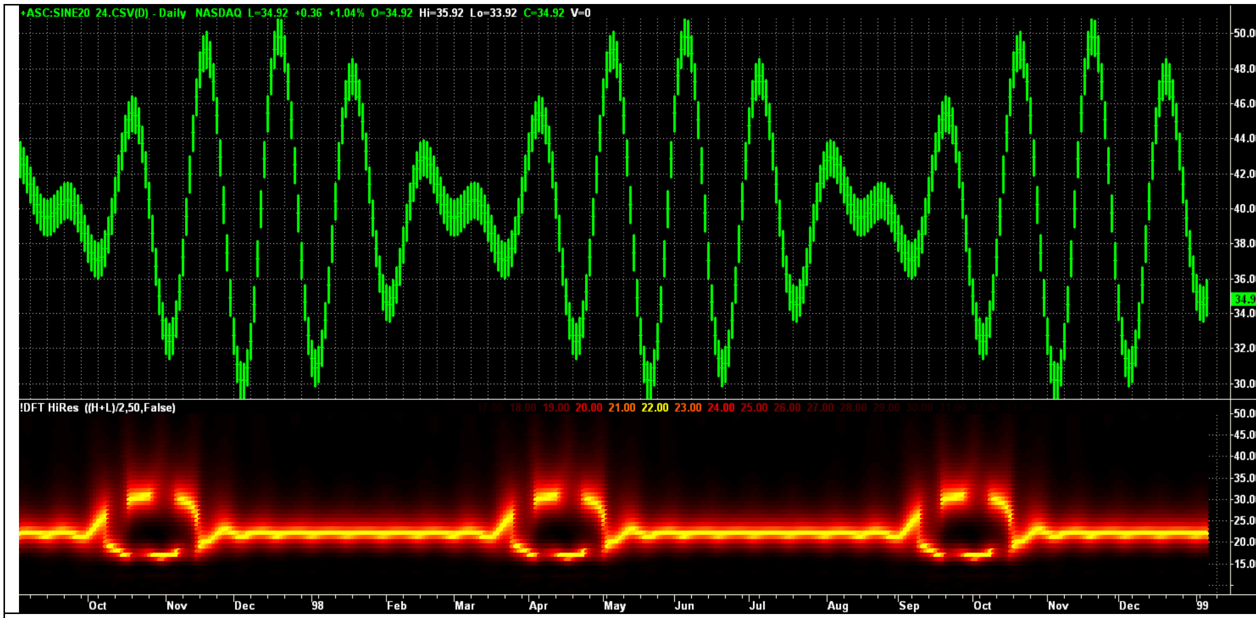

Figure 6. Transformed DFT Measurement of 20 & 24 Bar Cycles Cannot Isolate the Two Components

I also applied MESA8 to this same data, with the result displayed in Figure 6. Clearly, MESA8 does, in fact, have sufficient resolution to identify both cycles as be present.

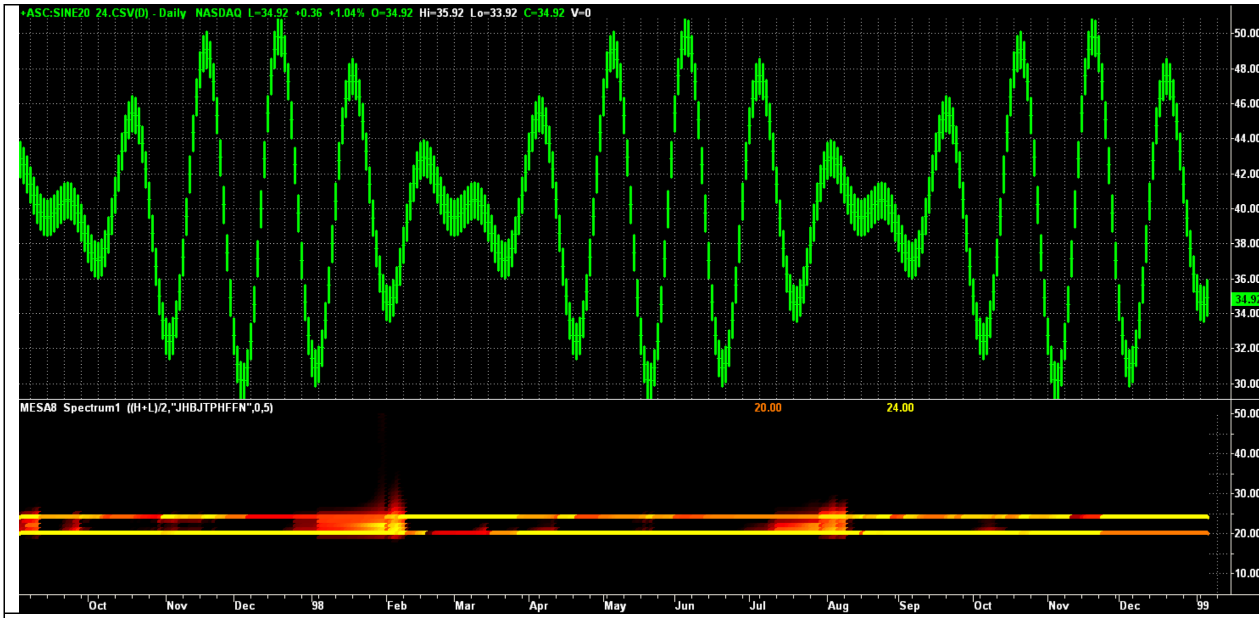

Figure 7. MESA8 Measurement of 20 & 24 Bar Cycles Has High Resolution, Clearly Identifying the Presence of both Components.

Although the Kay and Demeure transform improve the resolution of a DFT Spectral Estimate, I still have some problems using the transformed DFT approach without reservation. For example, the results generally track my more advanced techniques, but I am not convinced of its accuracy in low signal to noise environments. Further, it still takes a reasonably amount of historical data to make a measurement. That means it can be sluggish in response to rapid changes and mixed cycle periods within the observation window and get averaged together even when their periods are widely spaced.

Still, it is better than driving in the fog.

John Ehlers is a pioneer in the use of cycles and DSP techniques in technical analysis. He is the author of the MESA8 program, and www.eMiniZ.com and www.IndiceZ.com websites for trading.