

# **Fast Fourier Transforms (FFT)**

## **By Mark Patrick**

### **Background:**

In the 19<sup>th</sup> century Jean Baptiste Joseph Fourier created a method that breaks down any periodic function into an infinite series of sine and cosine curves. These sine and cosine curves are almost always much easier to deal with than the original curve due to their periodic, smooth, and differentiable qualities. However, when the Fourier Transform was applied to specific problems on early computers, the transform was found to be too time consuming. Programmers began working on new solutions. In 1965, two men named James Cooley and John Turkey worked together to produce the algorithm that we know as the Fast Fourier Transform.

### **Applications:**

Fast Fourier Transforms are very good at recreating price curves while reducing the high amount of noise that occurs in trading. Decreasing the noise of a price curve lets you look more at the general trends that have occurred, which will aid you in creating more effective trading strategies. FFT are also very good during cyclic markets since the price curve is broken down into sine and cosine components, both of which are cyclic. Fast Fourier Transforms are not as useful during trending markets because these sine and cosine curves have a tougher time dealing with non-cyclic markets.

### **Analysis:**

Any FFT analysis consists of a few basic steps. First, transform the curve into a series of sine and cosine curves. When the infinite series of sine and cosine curves is summed, the result is the original curve. This symmetric effect, named super positioning, is vital to the FFT algorithm.

For further analysis, break down the infinite series onto a spectrum. The spectrum graph consists of a period, measured in time on the X-axis and an amplitude, on the Y-axis. NeuroShell automatically hides this step so that it can properly filter and perform the inverse FFT and output the smoothed FFT indicator.

The next step in analysis of the spectrum graph would be to simplify the data. Because there are so many periods, ranging all the way out to infinity, there can be certain filters to filter out only the data that would be useful to your analysis. The simplest filter is the period filter, where you specify a range of periods that you would like to see more clearly. As the periods approach infinity, they become less and less useful for analysis because they almost always have little or no amplitude associated with them, meaning that they only have very minute effect on the overall spectrum. For example, a stock was seen to have a dominant cycle of 25 bars. While any complete FFT counts in every cycle between 1 and an infinite number of bars, we want to take a look at the cycles that would be most useful to trading. It would be safe to say that no cycle over

500 bars would have any noticeable effect on the inverse transform, and no cycle over 100 bars would have any significant effect on the inverse transform. Also, if you would like to eliminate the higher frequency noise in the price data, you could eliminate any cycles with a period less than 3 bars.

After the data has been filtered, the inverse FFT can be applied. The inverse FFT attempts to recreate the original curve with the infinite series of sine and cosine curves, excluding the certain periods that you have already filtered. In the example above, the price curve would be recreated automatically using only the periods within the range of 3 to 100 bars, rather than summing the whole infinite series of cycles.

### How NeuroShell Trader Handles FFT:

NeuroShell comes preloaded with ten FFT indicators. These indicators fall into two distinct categories, one category deals with the spectrum (a range of periods) and the other category deals with only one of the values on the spectrum (a single period). A period is defined as the number of bars it would take for a specific cycle to happen one time.

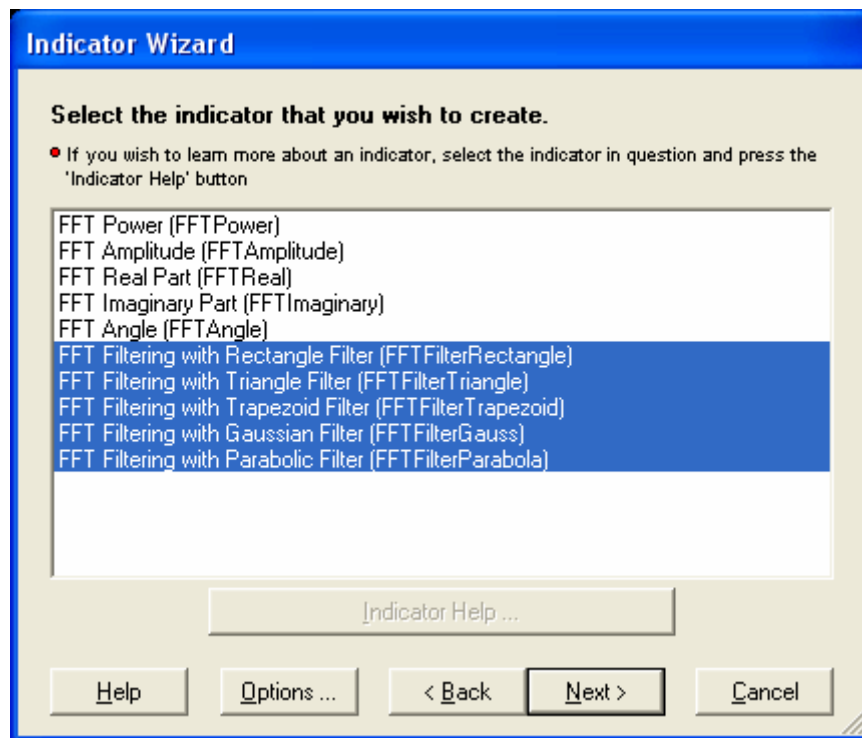


Figure 1: These highlighted FFT indicators included in the NeuroShell Trader work with a range of periods.

The category that deals with the entire spectrum has four major options. The first choice you need to make is which type of secondary filter you would like to use. There are five choices: Rectangle, Triangle, Trapezoid, Gaussian, and Parabolic. These secondary filters aim to filter out any extraneous spikes in amplitude of the period range

you have selected by superimposing a curve of selected shape over top of the period spectrum. Once you select a secondary filter, the next menu has a few more options.

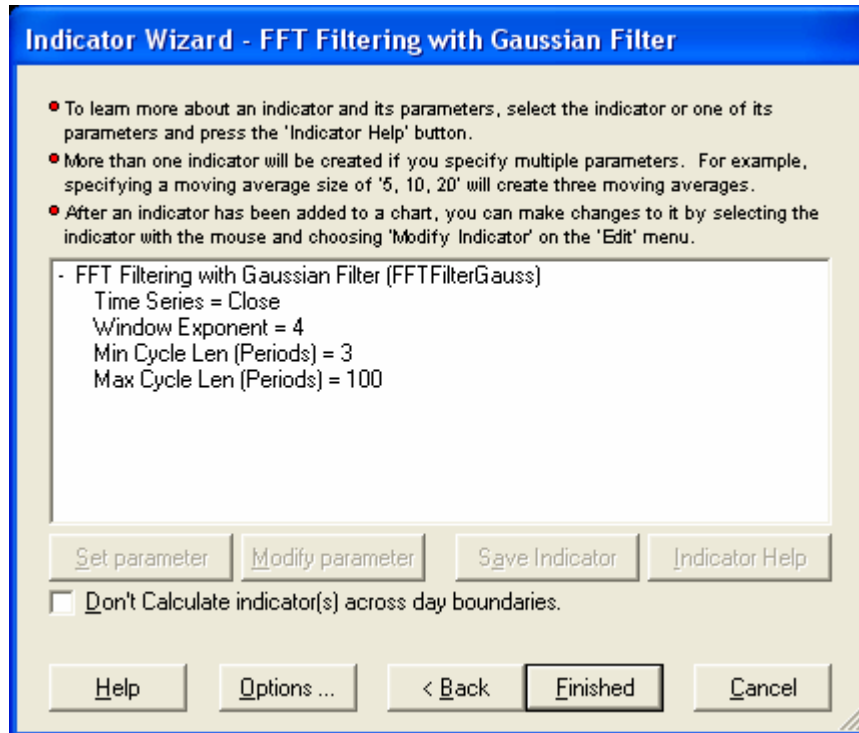


Figure 2: These are the customizable parameters after you select the secondary filter of the FFT; their meanings are outlined below.

The minimum and maximum cycle length are what is known as the primary filter, where you specify the period values that you would like to see. The primary filter helps eliminate very high frequency noise and very low frequency noise to better see the higher amplitude curves. For example, a sine curve with a period of one bar would be considered high frequency noise, a sine curve with a 50 bar period would be a higher amplitude period, and finally a sine curve with a 20,000 bar period will most likely have a negligible amplitude. To solve this issue, you may make the minimum period length 3 bars and the maximum period length 100 bars. When NeuroShell performs the inverse FFT algorithm, any periods not included in the range that you provided will not be reformed into the inverse FFT. NeuroShell automatically performs the inverse FFT after you select both filtering parameters while it is graphing the indicator.

Also, there is another option called the Window Exponent, which determines the number of bars you will be analyzing with the FFT. The FFT uses the Window Exponent as the exponent of base two to calculate the number of bars of which you will be taking the transform. For example, a Window Exponent of 4 would have a Window of 16 ( $2^4$ ) bars and a Window Exponent of 6 would have a window of 64 ( $2^6$ ) bars. The smaller the Window Exponent, the smaller the look back period of the indicator, as well as the smaller the amount of lag. The final option is the time series, where you can change the data series you will be transforming, the default here is the close data. However, since FFT indicators can also act as a smoothing indicator, you can take the FFT of any time series, you are not just limited to price curves.

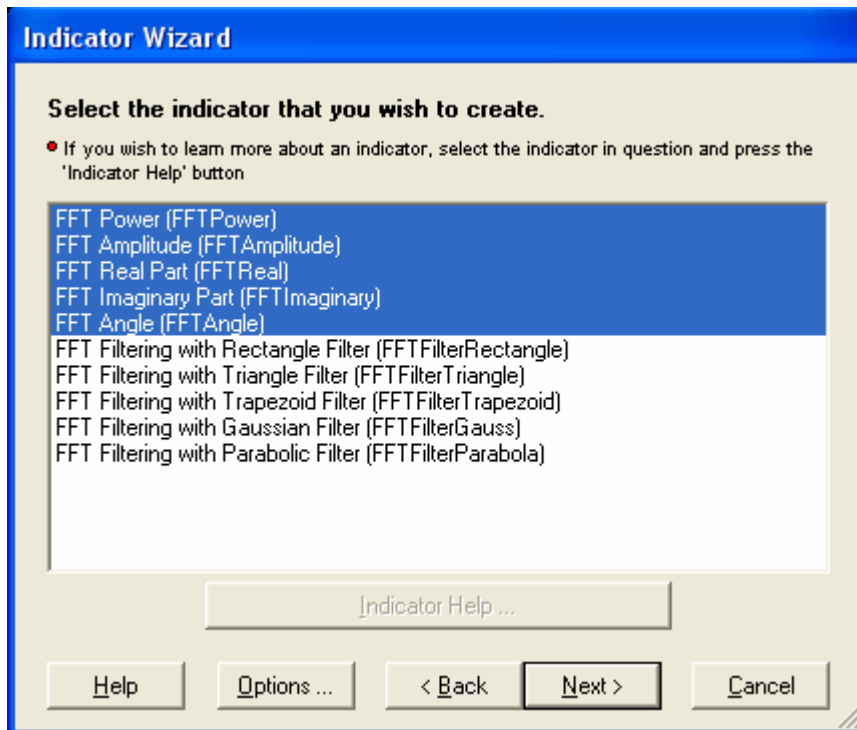


Figure 3: The highlighted FFT indicators work with a single period, yet the Power and Amplitude indicators will sum the Powers and Amplitudes respectively over a range of periods for you.

The formula for a specified period of a FFT falls along the lines of  $FFT=e^{i\Theta}$ , or  $FFT= A*\cos(\Theta)+iB*\sin(\Theta)$ . Thus, the coefficients of the sine curves are inherently imaginary. The second category, comprised of the first five indicators in the FFT indicator menu, of FFT indicators deal with only one specific period, and some of them deal with the breakdown of one specific period to sine and cosine amplitudes.

The indicators are designed to display the Power, Imaginary Amplitude, Real Amplitude, Total Amplitude, or the Angle of that specific period. However, to save you time, the Power and Amplitude indicators let you summate over a range of periods. The Real Amplitude indicator shows you the amplitude, A, of the corresponding cosine component to the period that you selected, while the Imaginary Amplitude indicator shows you the amplitude, B, of the corresponding sine component to the period that you selected. The Angle\*\* indicator is useful in showing which component, either sine or cosine, dominates the other for the given period. For example, an Angle near  $\pi/2$  would show almost no cosine amplitude, meaning, for the specified period, there is only sine amplitude. Also, an angle of 0 would mean that there would be no sine amplitude component, only cosine would have an amplitude at the specified period. Whereas the Real Part and Imaginary Part indicators show you the amplitudes of the cosine and sine functions, A and B, respectively for the specified period as defined by the formula  $FFT= A*\cos(\Theta)+iB*\sin(\Theta)$ . These three indicators have a more theoretical basis and are much more useful to engineers and physicists than to traders. However, feeding the outputs of the these FFT indicators as inputs of a Neural Network may be more advantageous. The Total Amplitude indicator uses the Pythagorean Theorem\*\*\* to add the Real and

Imaginary parts together to come up with a Total Amplitude. Since Power is correlated to the square of the Amplitude, the Power indicator squares the value of the Total Amplitude\*\*\*\* indicator. Amplitudes are useful in determining the effect one period has on the entire series. For example, a specific period with a low amplitude may not have a major effect on the series and may be eliminated using the primary filtering described above.

### Further Applications:

To master the cyclic markets you first must be able to differentiate between a trending market and a cyclic market. John Ehlers, in his MESA8 indicator package, does just that. Using the MESA8\_Momentum indicator, you can see whether markets are trending, or relatively cyclical, which would then merit use of the Fast Fourier Transform indicators. Another of his indicators, MESA8\_Period, graphically displays the period with the highest amplitude, which is usually referred to as the dominant cycle, for that given bar. With this indicator, you can see the changes in the dominant cycle over time.

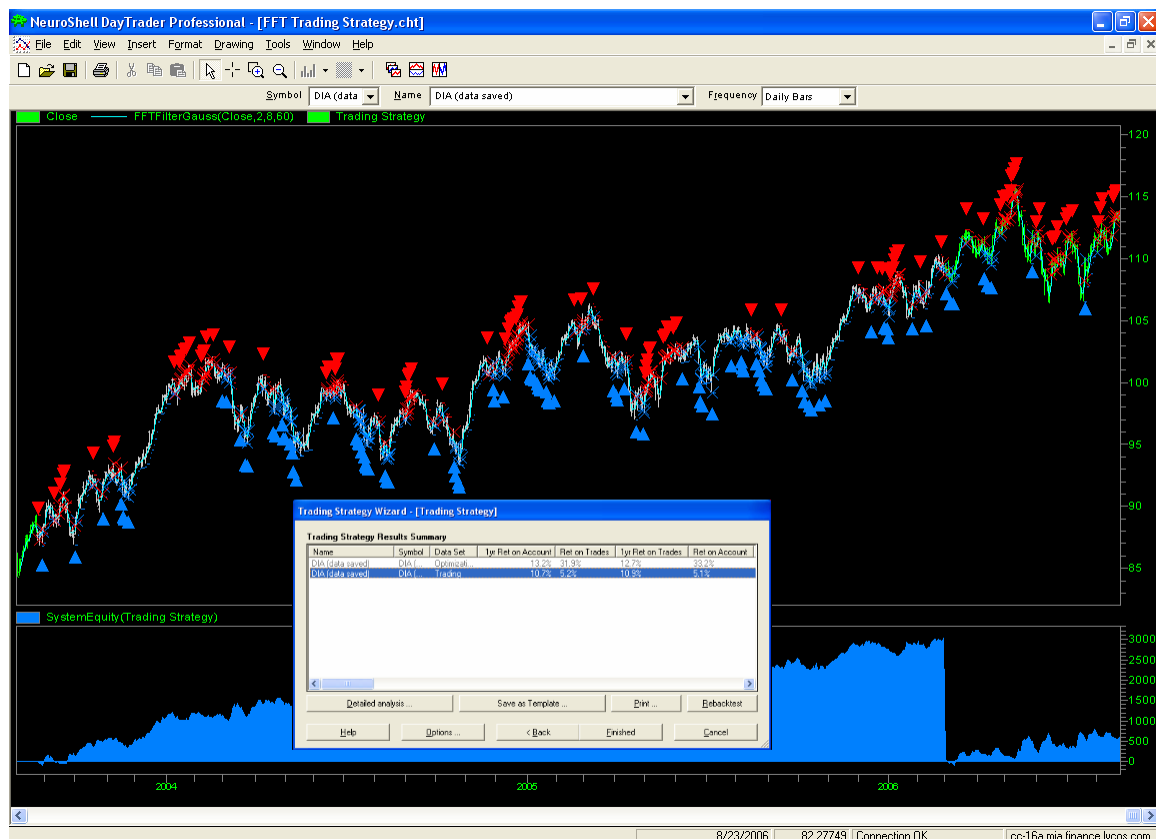


Figure 4: Above is a trading strategy that was created using FFT indicators. The associated chart is available for download at the bottom of the page. Using the FFT Gaussian indicator, with a higher Window Exponent lags the curve, as well as smoothes it, allowing us to create a crossover system with the close price and the FFT Gaussian indicator. Both the Long and Short entries have trailing stops of the lowest and highest price respectively in the last five bars.

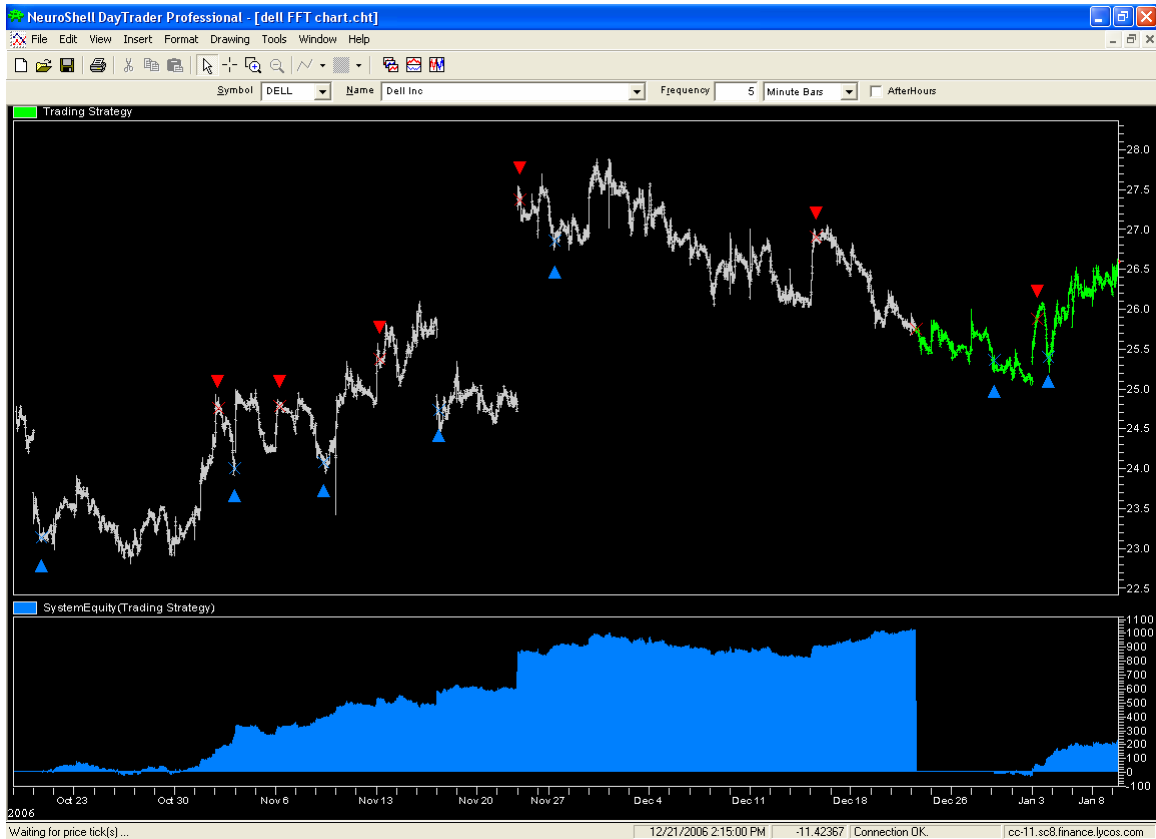


Figure 5: FOR DAY TRADER USERS ONLY. Above is a rule based trading strategy using the Dell stock on five minute bars. We used a FFT Gaussian indicator with a small Window Exponent to smooth the price curve with relatively little lag. Then, using moving averages of two differing look back periods, we constructed a moving average crossover system using the FFT Gaussian indicator as the input, rather than the close price. In addition, we added another rule to the trading strategy, a threshold based RSI signal. Where if the RSI of the FFT Gaussian indicator is below a certain threshold a buy signal was given, and if it was above a certain threshold, a sell signal was given. Buy orders were only placed when the RSI signal was to buy and the moving average of the FFT Gaussian indicator with the smaller period was larger than the moving average with the larger period. Likewise, sell orders were only placed when the RSI signal was to sell and the moving average of the FFT Gaussian indicator with the larger period was larger than the moving average with the smaller period.

### Side notes for more advanced users:

\*\*The Angle indicator uses the function of arctangent of the Imaginary Amplitude divided by the Real Amplitude. The resulting angle is measured in Radians. The arctangent function is defined over the interval  $\pi/2 < \Theta < \pi/2$ .

\*\*\*The Pythagorean Theorem describes the behavior of the relationship between the sides of a right triangle. It can be applied to this situation in adding Real and Imaginary parts because they are always perpendicular to each other. Also, in this case, the Pythagorean Theorem is used to find the magnitude of the resultant of the vectors, which acts like the hypotenuse of a right triangle.

\*\*\*\*Note that the sum of the amplitudes of all periods will be greater than the value of the price at any given point. This mathematical oddity occurs because amplitudes are always positive. Even if two cycles are destructively interfering with each other, their amplitudes will still be added in the case of the Amplitude indicator.