The End Point Fast Fourier Transform

In an article entitled “The Discrete Fourier Transform Illusion” the misuses of the mathematical technique called the Fourier Transform were demonstrated as applied to the S&P500 index. There it was shown how fitting the Fourier Transform to the S&P500 index daily series produced a perfect curve fit on past data giving the illusion that this technique would predict the major turning points of the S&P500 index. However, in that article it was demonstrated that when the Fourier Transform was examined on a day by day walk forward basis this seemingly wondrous predictive capability disappeared.

Here it will shown how to use Fourier Transform, using the computational algorithm called the Fast Fourier Transform (FFT), on a walk forward basis on the S&P500 Mar/00 futures using 5 minute bars.

The End Point FFT

In order to avoid the past data curve fit illusion, an indicator that walks forward one day at a time will be created. This indicator will calculate a noise filtered FFT curve at each price bar but only save the last point, or end point, of the curve on the bar that it is calculated. All the generated end points will then be connected to produce a curve that matches what would have been seen if the noise filtered FFT was performed on the last bar as each last bar is formed marching forward in time.

As an example supposed the FFT was calculated using daily data from 7/20/98 to 7/20/99. The FFT of the daily price data would be noise filtered and transformed back into a filtered daily time series. The filtered curve point of 7/20/99 would be saved and the data window would slide forward one day from 7/21/98 to 7/21/99. A new filtered daily time series would be computed as above and from this new data window the filtered FFT curve point of 7/21/99 would be saved. We would keep sliding the data window forward one day at a time, calculate a new noise filtered curve and save the filtered curve end point. Then all these saved endpoints would be connected to create a new curve which is dubbed the End Point FFT or EPFFT for short. The EPFFT represents the curve value that would be observed in real time on the last bar if this procedure was performed on each bar as the bar occurred.

EPFFT Construction Details

Unfortunately constructing the noise filtered FFT of a price data series is not quite as simple as just taking 256 closing prices, and directly plugging them into a FFT algorithm.

The mathematics of the Discrete Fourier Transform assumes that the time-domain sample is periodic and that it has captured an integral number of periods. This means that the DFT assumes the end of the sampled series implicitly wraps around to the beginning to start all over again. Thus for 256 sampled data points the DFT assumes that these 256 sampled data points repeat every 256 samples. With real data series this is seldom the case and this creates what is called a wraparound effect in the frequency domain. The wraparound effect creates a lot of distortion on the ends of the time series when the noise filtered FFT is transformed back. Unfortunately the end point is the very point we wish to estimate. While we can do nothing about the wrap around violation, we can significantly lessen it's effects by what is called zero padding.

The DFT assumes the time domain sample is periodic and repeats. Suppose the time sample starts at 400 and wiggles and wags for 256 data samples ending at the value of 600. The DFT assumes that the time series starts at zero, suddenly jumps to 400, goes to 600 and suddenly jumps down to zero again and then repeats. The DFT must create all kinds of different frequencies in the frequency domain to match this type of behavior. These false frequencies created to match the jumps and the high average price completely swamp the amplitudes of any real frequencies making them look like noise. Fortunately this effect can be almost eliminated by a simple technique called end point flattening.

The calculation of end point flattening coefficients is simple. If x(1) represents the first price in the sampled data series, x(n) represent the last point in the data series and y(i) equal to the new endpoint flattened series then:

$$\mathbf{a = x(1) \quad b=(x(n)-x(1))/(n-1)}$$

$$\mathbf{y(i) = x(i) - [a+ b*(i-1)] \quad \text{for } i=1 \text{ to } n \quad (1)}$$

We can see that when i=1 then y(1)=0 and when i=n then y(n) =0. What we've done is subtract the beginning value of the time series to make the first value equal to zero and then rotate the rest of the time series such that the end point is now zero. This technique reduces the endpoint distortion but introduces a low frequency artifact into the Fourier Frequency spectrum.

EPFFT Curve Construction

For this article a sliding time-bar window of 512 5min bars of the SP Mar/00 futures from 11/30/99 to 12/31/99 will be used. The first time window will be the 512 price bars from 11/30/99 to 12/06/99.

Step 1 Take the Log_e of the 512 prices to minimize the exponential trend movements of the S&P500. End flatten these 512 log prices using equation (1) above.

Step 2 Take the FFT of these 512 points to create 512 frequency domain complex numbers $\mathbf{f_i}$.

Step 3 Use a Low Pass filter that zeros out the frequencies whose magnitudes are less than 0.15*Fmax. Fmax is the maximum frequency amplitude of the spectrum.

Step 4 Do an inverse FFT on the noise filtered spectrum and save the last 2 points. Although it's not obvious yet, we will need the noise filtered point one sample before the endpoint.

Add back the end point flattening of **Step 1** and taking the exponential value to reverse the log_e

Step 5 Save the filtered end point and the point before. Call the endpoint y(k) and the point before y1(k) where k is the denotes the order of the sliding window. That is the first sliding window k=1, the second, k=2, etc. Slide the 512 bar data window forward one bar, and repeat steps 1 through 4.

When the data window is slide forward one bar at a time a new data sample is added to the end and the data sample at the beginning is subtracted. This adding and subtracting causes the end point flattening coefficients and the power in the frequency spectrum to jump around creating distortion and jitter in the calculation of the noise filtered end point. This random jumping of the FFT endpoints as the data window slides forward in time adds a random jump to FFT end point curve. Fortunately this jumping can be minimized by creating a new end point curve from the two saved noise filtered end points, y(k) and y1(k), above in step 5. Since turning points are of interest rather than magnitude then in **step 5** a new variable will be created called **dsumEP** where

dsumEP(k) = dsumEP(k-1) + y(k)-y1(k)

This new curve **dsumEP(k)** is the sum of all the changes in the individual y(k)'s from their noise filtered FFT value one sample before. This change series minimizes the magnitude jump problem creating a fairly smooth EPFFT curve.

The EPFFT System Defined

Even though dsumEP is a fairly smooth curve as seen in Figure 5, it still has a number of short term wiggles preventing us from simply going long when the curve turns up and going short when the curve turns down. To create a system, we will use a simple curve following technique.

Buy Rule:

- **IF dsumEP** has moved up by more than the point amount of *pntup* from the lowest low recorded in **dsumEP** while short then buy the SP Mar/00 futures at the market..

Sell Rule:

- **IF dsumEP** has moved down by more than the point amount *pntdn* from the highest high recorded in **dsumEP** while long then sell the SP Mar/00 futures at the market.

Exit Rule:

- Exit on the close.

Walk Forward Optimization

Walk forward optimization will be used here because of the changing nature of the intraday S&P futures market. Intraday price dynamics are constantly changing due to current economic surprises, events and trader sentiment. Also the time of year changes the nature of intraday markets, such as the seasons, holidays, vacation time, etc. As such, optimizations on intraday data performed 3 months ago may no longer be representative of today's intraday price dynamics.

The walk forward procedure will be applied as follows. A test period of 4 weeks of the S&P Mar/00 5 minute bar data, November 30th, 1999 through December 30th, 1999, is chosen and system parameter values are found through optimization on this intraday data segment. The parameter values found are then applied to the out-of-sample 5 minute intraday bar data following the test segment which in this case is January 3rd, 2000 to January 7th, 2000. December

31st, 1999 was a shorted day with the Y2K scare so it was excluded from the out-of-sample results.

Why a 4 week intraday data test segment?   There is no correct ratio of test data needed to produce good one week intraday out-of-sample results. By experimenting with different window lengths, the four to one ratio seemed to work well. In walk forward testing, enough data is needed to model most of the price dynamics that will be encountered in the out-of-sample segment, but not so much data that when the price dynamics start to change they are swamped by the weight of distant past data price dynamics that no longer are valid. An important unspoken point in walk forward testing is that if you can not get good results in the out-of-sample segments, then the price dynamics cannot be modeled with the system . This means that real time performance will be random using the model. Traders observe this type of random performance (that is it looks great on paper but falls apart in real time) when trying systems based on curve fitting or anecdotal “proof” (looking at 3 or 4 successful cases only) without any out-of-sample testing .

Finding The System Parameters Using Walk Forward Optimization

There are two system parameters to find   *pntup, and pntdn*. The best parameters will be defined as those values that give the best Net Profits with the maximum winning bars, minimum losing bars, minimum drawdown, minimum largest losing trades. In addition, the results should be stable, e.g. the profits, wins, and drawdowns should not change by much as the parameters move by a small amount away from their optimum values. Also in choosing the “best” parameters, only those parameters sets whose maximum consecutive losses were 4 or less were considered. Optimization is defined as the search for the parameter values that give the best results as defined above. It should be noted that in this stage of system development, the only thing indicated by the optimum values that are found in the test portion is that the data has been *curve fitted* as best it can with this system. Without further testing on out-of-sample data there is no way to tell if the system will work in the future.

It is not well known, but almost any real time series or even a random time series defined over a fixed number of bars can be curve fitted rather easily. The performance results and the statistical measurements that validate this performance of the curve fit will look excellent giving the false illusion of future profitability. However, the truth is that these excellent performance and associated statistics on the test section in no way validate how the system will perform on data it has not been optimized on. Only out-of-sample testing, that is testing on data the parameters were not derived on, can determine if the parameters found in the test section have captured the price dynamics. For instance in the End Point FFT process the error minimization forces the generated curve to fit the past data like a glove. It’s almost impossible not to get an excellent fit with excellent statistical results. Unfortunately, this excellent fit in no way implies that the system will perform equally well on out-of-sample data, it just tells us we have a very good curve fit.

Results

Figure 1 presents a table of the test window optimum parameters for the S&P Sep/99 5min data series.

Start Date	End Date	Pntup	Pntdn
11/30/1999	12/30/1999	1.0	3.0

Figure 1 Optimum Parameter Values For Test Data Segment

Figures 2 presents the performance summary of the test segment using the optimum parameters shown in Figure 1.

Figure 3 presents the performance summary of the out-of-sample data segment from 01/03/2000 to 01/07/2000. This performance represents what would have happened in *real time* if one used the parameters found in the test section. Slippage, and commissions are not included.

Figure 4 presents a trade by trade summary from 1/3/2000 to 1/7/2000. Note only the out of sample trades are presented here for the in sample trades were generated by the curve fit.

Figures 5A through 5B present the 5 minute bar charts of the S&P Mar/00 futures with the EPFFT Curve and all the buy and sell signals from the trade by trade summary of Figure 4 indicated on the charts.

Discussion of System Performance

As can be observed from the test sample Performance summary in Figure 2 and the out-of-sample performance summary of Figure 3, the out-of-sample performance was better than the test sample performance. This better performance in the out-of-sample section was mainly due to the $11,125 gain on 1/7/2000 and may have been due to chance. The out-of-sample gains do indicate that 4 weeks of test data was enough to capture the intraday price dynamics of the S&P Mar/00 contract for one week into the future..

Observing the out-of-sample trade by trade summary of Figure 4, we can see that the system did equally well on longs and shorts. This is a good sign showing that neither longs nor shorts were favored in the current bull market. The average trade (win & loss) was $224 in the test section and $1236 in the out-of-sample section(taking out the big $11,125 gain on 1/7). This indicates stability in the parameter selection. The test section had a rather low average trade profit. This was because the test section was a rather whipsawey market with a number of big jumps on the open. Also the last week of December 1999 had a very small range. What is interesting about the out-of-sample performance is that the out-of-sample section used the parameters derived in the whippy test section but performed outstandingly in the big down and up trends during the first week of January 2000.

In order to use this system in real time trading, at least ten to twenty more test and out-of-sample windows from the past would have to be examined to gain confidence that the results above were not due to pure chance.

References:

Meyers, Dennis , “The Discrete Fourier Transform Illusion”, *Stocks & Commodities*, Volume 17: April.1999

Info on Dennis Meyers

Dennis Meyers (info@MeyersAnalytics.com.) has a doctorate in applied mathematics in engineering. He is a private trader, and president of Meyers Analytics (www.MeyersAnalytics.com). His firm specializes in Financial Engineering consulting for financial institutions and developing publicly available analytical software for traders.

Figure 2 Test Window Performance Summary for S&P Mar/00 EPFFT System 11/30/99 - 12/30/99

```
EPFFT System- S&P Mar/00-5 min   11/30/1999 - 12/30/1999

       Performance Summary:  All Trades

Total net profit      $  6500.000    Open position P/L     $     0.000
Gross profit          $ 21450.000    Gross loss            $-14950.000

Total # of trades          29        Percent profitable         45%
Number winning trades      13        Number losing trades       16

Largest winning trade  $  5575.000    Largest losing trade  $ -2100.000
Average winning trade  $  1650.000    Average losing trade  $  -934.375
Ratio avg win/avg loss       1.766    Avg trade(win & loss) $   224.138

Max consec. winners         5        Max consec. losers          5
Avg # bars in winners      52        Avg # bars in losers       33

Max intraday drawdown  $ -5425.000
Profit factor                1.435    Max # contracts held         1
Account size required  $  5425.000    Return on account          120%
```

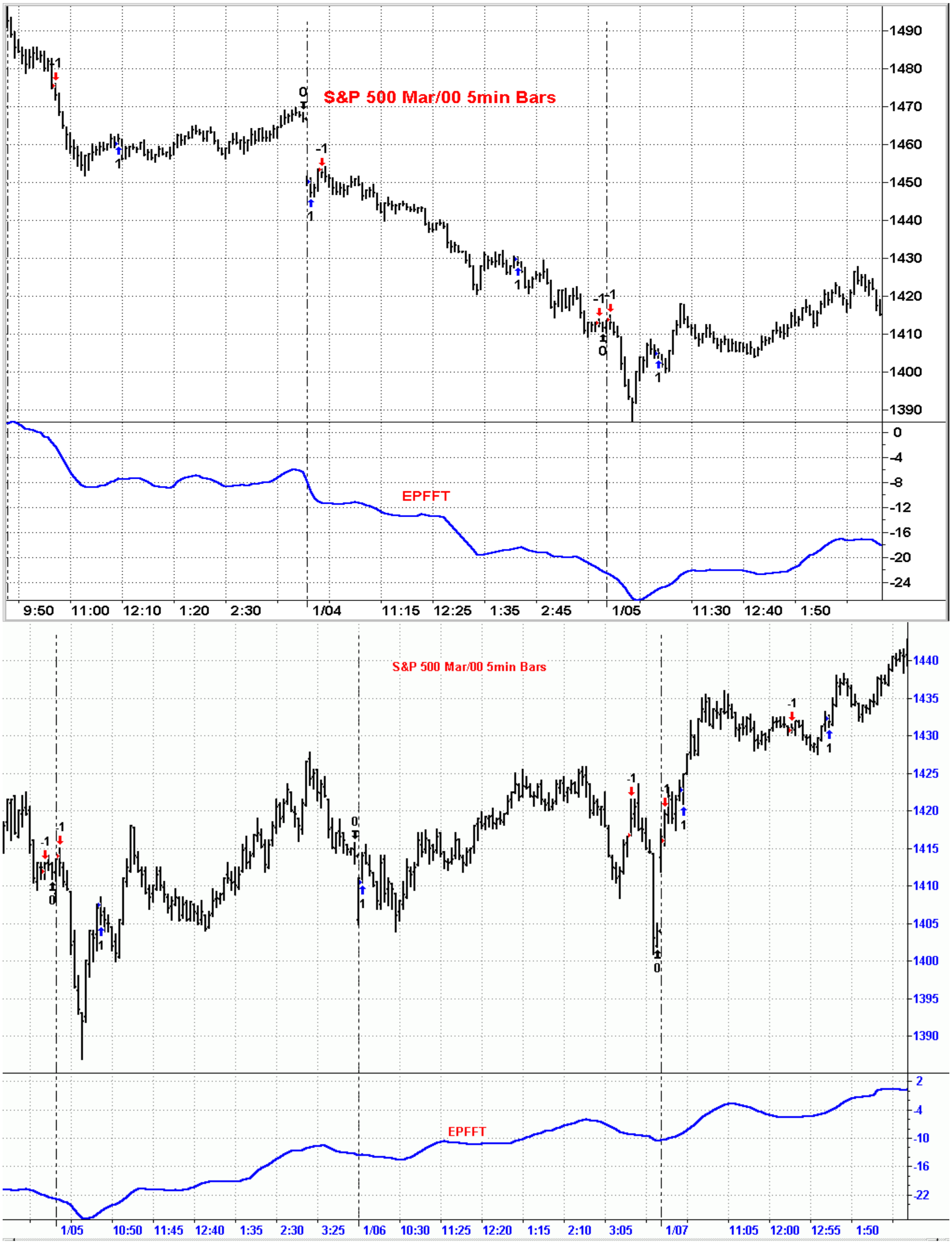
Figure 3 Out-Of-Sample Performance Summary for S&P Mar/00 EPFFT System 1/3/2000 – 1/7/2000

```
EPFFT System- S&P Mar/00-5 min   1/3/2000 – 1/7/2000

       Performance Summary:  All Trades

Total net profit      $ 22250.000    Open position P/L     $     0.000
Gross profit          $ 28075.000    Gross loss            $ -5825.000

Total # of trades          10        Percent profitable         80%
Number winning trades       8        Number losing trades        2

Largest winning trade  $ 11125.000    Largest losing trade  $ -4200.000
Average winning trade  $  3509.375    Average losing trade  $ -2912.500
Ratio avg win/avg loss       1.205    Avg trade(win & loss) $  2225.000

Max consec. winners         4        Max consec. losers          1
Avg # bars in winners      35        Avg # bars in losers       51

Max intraday drawdown  $ -5075.000
Profit factor                4.820    Max # contracts held         1
Account size required  $  5075.000    Return on account          438%
```

FIGURE 4 Out-Of-Sample Trade By Trade Summary
S&P Mar/00 5min EPFFT System 01/03/2000 - 01/07/2000

Entry	Entry		Entry	Exit	Exit	Exit	Bars	Trade	Trade		Trade	
Date	Time		Price	Date	Time	Price	InTrd	$P&L	Max$Pft	Time	Max$DD	Time
1/3/00	1040	Sell	1475.5	1/3/00	1205	1460.2	17	3825	5250	1115	0	1040
1/3/00	1205	Buy	1460.2	1/3/00	1615	1466.8	50	1650	2025	1605	-1000	1210
1/4/00	940	Buy	1450.3	1/4/00	955	1453.4	3	775	800	950	-750	940
1/4/00	955	Sell	1453.4	1/4/00	1420	1429.8	53	5900	7975	1325	0	955
1/4/00	1420	Buy	1429.8	1/4/00	1610	1413	22	-4200	0	1420	-4700	1550
1/4/00	1610	Sell	1413	1/4/00	1615	1411.8	1	300	300	1615	0	1610
1/5/00	940	Sell	1414	1/5/00	1045	1405	13	2250	5500	1010	0	940
1/5/00	1045	Buy	1405	1/5/00	1615	1414	66	2250	5375	1510	-1000	1055
1/6/00	940	Buy	1410.5	1/6/00	1615	1404	79	-1625	3625	1430	-1875	1610
1/7/00	940	Buy	1416	1/7/00	1615	1460.5	79	11125	11125	1615	0	940

FIGURE 5A S&P Mar/00 5min EPFFT System 01/03/2000 - 01/07/2000


S&P 500 Mar/00 5min Bars
EPFFT
9:50
11:00
12:10
1:20
2:30
1/04
11:15
12:25
1:35
2:45
1/05
11:30
12:40
1:50
S&P 500 Mar/00 5min Bars
EPFFT
1/05
10:50
11:45
12:40
1:35
2:30
3:25
1/06
10:30
11:25
12:20
1:15
2:10
3:05
1/07
11:05
12:00
12:55
1:50