Statistical Tests and Displays in *QuanTek*

(Revised October 19, 2017)

Robert Murray, Ph.D.
Omicron Research Institute

1.0 Contents

1.1 Random Walk & Efficient Markets

According to the *Random Walk* model, stock price returns (the changes in price over a given time period, such as from one closing price to the next) are supposed to be independent, uncorrelated random variables. (More precisely, it is the *logarithmic* price returns that are usually considered. These are postulated to be Gaussian random variables.) Then the logarithmic prices, which are the sum of these independent price returns, follow a *stochastic process* called the **Random Walk**. The main consequence of the Random Walk hypothesis is that future returns are independent (and hence uncorrelated) with the past prices (or any other financial data, such as fundamental data). So, theoretically no function of past data can be used to predict future price returns. This is a statement of the **Efficient Market Hypothesis**, of which the **Random Walk** model is a special case.

If the market were perfectly efficient, then there would be no point to short-term trading. On the average, the expected return from **short-term trading** would be zero, relative to a **buy-and-hold** strategy. If the **Random Walk** process is one with drift, corresponding to the secular upward trend of the stock market, then the buy-and-hold strategy would give an overall average **return** over a long holding period equal to the secular trend. This is presumably a reward for the **risk** inherent in stock investing, which is measured by the *variance* or *standard deviation* of the **Random Walk** over time. But short-term trading would only increase the **risk**, with no corresponding increase in **expected returns** over time. Thus it would be just like gambling, except that the expected return (over buy-and-hold) would be zero (rather than a loss, as with most gambling).

However, hardly anybody believes that the market is truly **efficient**. There are many people interested in short-term trading, and many others who are prudent to buy and sell securities over longer term holding periods, as the situation changes and different securities look more promising (based on past information). A simple, rough argument indicates that *the market can never be truly efficient*. If the market were perfectly efficient, then there would be no reward for short-term trading (or longer-term trading either), so people would stop trading. But it is precisely the trading activities, on *all time scales*, that keep the market efficient. Hence when the trading stops, inefficiencies would immediately be created, which would induce people to start trading again because they would then be able to make a profit. So the conclusion is that people trade to the extent that they can still make a profit, so the efficiency of the market is dictated by the ability of the best traders to still be able to make a profit (at the expense of the less knowledgeable traders). So we expect inefficiencies to exist at a level that the most sophisticated traders are just able to find and take advantage of them (in a best case scenario). At the present time, the market is *almost* efficient, but it can never be *perfectly* efficient. Profitable trading opportunities will always exist for the most sophisticated traders.

1.2 Stochastic Noise & Random Shocks

The **Random Walk** model may be thought of as a model in which each price movement is an independent *random shock*. Such a *random shock* is presumably the result of some business development or news input regarding the security, at least for the larger shocks. However, probably a more realistic model of stock price behavior is that it is due to **random shocks** occurring at infrequent intervals, and in between the shocks the price action is due to investor reaction to these shocks. This investor reaction is not instantaneous, in the real world, so the market is not perfectly efficient. The investors react to the shocks and the present state of the market with some finite time delay, which is of the order of the investment horizon of that investor. Also, many investors do not know how to properly interpret the present condition of the market, so they over-react and cause prices to swing above or below their “fair value”. This combination of inefficiencies should cause some sort of dynamical behavior of asset prices in response to the *shocks* due to external influences, such as the state of the company itself or of the overall economy, or political events. So, we have a set of shocks, with large shocks occurring at infrequent intervals, and smaller shocks occurring more frequently, according to some power spectrum, say, and a dynamical reaction to the shocks, which is delayed in time according to the spectrum of time horizons of all the investors. So the result is a spectrum of **unpredictable random shocks**, and of **predictable dynamical responses** to those shocks. It is these dynamical responses that **Technical Analysis** hopes to capitalize on by means of various indicators. But the point is that, due to the finite response time of investors, the deterministic part of the price patterns are, to some extent, smooth and slowly varying. At least, that is our hypothesis, assuming the use of end-of-day data. There is additional correlation in intra-day tick data for time scales shorter than, say, 20 minutes, but we are only making use of end-of-day data here. For a partial theory of correlation in price tick data, see The Econometrics of Financial Markets (1997) by Campbell, Lo, & MacKinlay [CLM].

Hence we may postulate a model for stock price action. It consists of a **deterministic** part, which can be predicted (in principle, if not in practice), which is smooth and slowly varying, and hence consists of the lower-frequency Fourier components of the returns process. To this is added a **random** part, which may be modeled as **stochastic white noise**, with a constant spectrum. Thus most of the high-frequency variation of prices is random, stochastic noise with very little predictive power. (However, an exception to this is the apparent anti-correlation of returns over time intervals of a few days.) In order to uncover the predictable, deterministic part, it is necessary to employ **smoothing** to filter out the high-frequency components. Otherwise, the small correlations in the low-frequency deterministic part are completely drowned out in the high-frequency noise and cannot be seen. This is probably why it has been found so many times that the stock price data are statistically a **Random Walk**, and no clear deviations from the **Random Walk** can be seen by the classical statistical tests. After smoothing the data, however, we do find some clear indications of usable correlations, although it should be emphasized that these are hardly ever very far above the level of the stochastic noise.

1.3 Measuring Correlation in Data

In order to find **trading rules** that work, we must find certain functions of the past price data (and/or perhaps other financial data such as fundamental data) that have a non-zero **correlation** with **future returns**. (See the Appendix for the definition of **correlation**.) As we have stated, the **Random Walk** model states that this correlation should be zero. We can construct various functions and measure their correlation with future returns, or more precisely, we can measure the **sample correlation**. The sample correlation is an *estimate* of the actual correlation, based on a finite sample of data. The true correlation can only be determined in a hypothetical stochastic system in which there is an *infinite amount of data* available, and the stochastic process is **second-order stationary**, meaning that the correlation is constant for the whole data set. And here is a major problem regarding financial data: There is almost never a very large data set to work with, and within this data set it is almost certain that the stochastic process is **non-stationary**. So the measured correlation within one block of data will (probably) be different from that within other blocks of data in the same data set. Furthermore, within a finite data set, the *sample correlation* is itself subject to a statistical uncertainty. A totally random data set can yield a measured value of the sample correlation, which is non-zero, just because of random statistical fluctuations. The standard error for these fluctuations, for the usual Linear or Pearson's R correlation, is given by $1/\sqrt{N}$, where N is the number of data points in the set. (The standard error is slightly smaller for the *robust* correlation methods.) So, for a set of returns 100 days long, the standard error of the sample correlation for these returns is 10%, which would be a very sizable correlation if it existed. For a data set 1024 days long, which is the usual length of the data set that we work with, the standard error is 3.125%, which would be a small but non-negligible correlation if real. Furthermore, there are indications that **long-range correlations** only extend to a maximum of 1024 data days, or four years [EP1, EP2]. So, the conclusion is that any correlations that exist in the data, are likely to be "down in the statistical noise" and of the same order of magnitude as the statistical uncertainty of the sample correlations. Nevertheless, these small correlations, if real, can lead to very sizable returns from short-term trading.

As an example, suppose we find a technical indicator that has a 5% correlation with the 1-day future returns. Suppose the daily volatility is 2% (r.m.s. value of daily returns). Then, setting the daily trading position (*trading rules*) proportional to the technical indicator, the expected daily gain is the product of the correlation times the volatility, or 0.1%. Assuming 256 trading days per year, this leads to a simple annual gain from short-term trading of 25.6% and a compounded annual gain of 29.2% (over buy-and-hold), which most people would regard as excellent! However, by most standards the 5% correlation, given a standard error of 3.125%, would not even be regarded as **statistically significant**. The conclusion is that if we want to find trading rules that work, we have to search for correlations that are barely above the statistical "noise" level, and as a result we must also accept that the standard deviation of the gains (from short-term trading) will inevitably be of the same order of magnitude as the gains themselves. Nevertheless, if the short-term trading is done within the setting of an overall portfolio strategy, the standard deviation for short-term trading for the whole portfolio can be reduced while the returns remain the same. In this case the standard deviation of the returns will be reduced roughly by a factor $1/\sqrt{N}$, where now N is the number of securities in the portfolio. Of course, to get this $1/\sqrt{N}$ reduction in the standard deviation, it is necessary to do N times as much work!

Regarding the *statistical significance* of the correlation, the usual interpretation is that a correlation greater than two standard errors (from zero) is regarded as *significant*. A correlation this large, at least 6.25% in the example above, is achieved only 4.6% of the time by pure chance alone. (This corresponds to a 4.6% **significance level**.) So, we say that this correlation is *significant* at the 95.4% **confidence level**, because there is a 95.4% chance that this correlation is *not* due to chance alone. (We are calling the *confidence level* that quantity which is 100% minus the *significance level*.) Theoretically, when estimating the "true" correlation by means of the sample correlation, the measured sample correlation will itself be a random variable with a Gaussian distribution of values. The standard error of this distribution is $1/\sqrt{N}$ as stated above, for a sample size N. Thus, if there is no actual correlation at all, then the measured values of the correlation will be distributed around zero, with a standard error $1/\sqrt{N}$. These values will lie within one standard error of zero 68.3% of the time, within two standard errors of zero 95.4% of the time, and within three standard errors of zero 99.7% of the time [SN]. So, if the measured correlation is not at least two standard errors away from zero, it is usually regarded as not **statistically significant**. However, this does *not* mean that if the measured correlation is within two standard errors of zero, then it is necessarily not a real correlation. All it means is that the measured correlation is *consistent* with zero correlation (to the 4.6% significance level). Most of the correlation we measure, at the "peaks" in the **Correlation Test** display in ***QuanTek***, are actually more than two standard errors away from zero, so they can be regarded as *significant*. However, we prefer the following interpretation, which seems more reasonable: The measured correlation represents the *mean* or **expected value** of the actual correlation, and this value is *uncertain* by an amount given by the **standard error**, $1/\sqrt{N}$. In this way we are not forced to ignore measured correlations that are within two standard errors of zero, and then "define" them to be zero. We regard the measured correlations to be the *most likely value* of the actual correlations, subject to a rather wide uncertainty given by $1/\sqrt{N}$. If Edgar Peters (1991, 1994) [EP1, EP2] is correct and the correlations do not persist longer than 1024 days or so, then we cannot reduce this statistical uncertainty any lower than about 3% by taking a larger data set, so there is never any way to conclusively separate the correlations we are seeking from the stochastic uncertainty of the sample correlation measurement. Nevertheless, these correlations, provided they are really there (which they *seem* to be), can still be used (we hope) to construct profitable (over the long term) short-term trading rules.

The ultimate point is that there is no "**Law of Large Numbers**", or mathematical limit as $N \to \infty$, that we can take in order to prove conclusively the existence of correlation, or measure the sample correlation to arbitrarily high confidence levels. This limit might be approximated by finding some trading rule, and testing it on a whole portfolio of stocks over a long period of time, say 2048 days. In this way, we may finally be able to find an unambiguous signal for a highly statistically significant correlation, and the portfolio ensemble then plays the role of the very large **statistical ensemble**. But such a calculation might take hours or days to perform, and I have not yet attempted such long calculations. In the meantime, it is still necessary to apply a certain amount of intuition in deciding which correlations are meaningful and which are just stochastic noise. (Having said this, I should add that the ***QuanTek*** program yields some rather clear signals for correlations between certain technical indicators and future returns, which certainly do not *look like* stochastic noise. But there is no statistical test that can prove *conclusively* that they are real correlations. Without an infinite data set, or at least a very large one, it is *impossible* to prove *anything* conclusively from Statistics.)

1.4 Definition of Technical Indicators

The usual definition of a **technical indicator** is some function of the past price data, which "signals" a buy or sell point. As a prototype, one of the most commonly used technical indicators is a combination of two (exponential, say) moving averages, one with a longer time scale than the other. When the shorter MA crosses the longer MA moving upward, this is a buy signal, and when the shorter MA crosses the longer MA moving downward, this is a sell signal. The expectation is that as long as the shorter MA is above the longer MA, the prices will be in an up-trend, and as long as the shorter MA is below the longer MA, the prices will be in a down-trend [Pr]. (Evidently there is an assumption here that the prices will be in one of two modes, either bull or bear market, and that these modes will last much longer than the time scale of the moving averages themselves.) Equivalently, we can form an *oscillator* from the two MA's, by subtracting the longer one from the shorter one (assuming logarithmic price data). This is a logarithmic version of an oscillator called the **Moving Average Convergence-Divergence (MACD)**, in which the ratio of two exponential MA's of the price data is taken [Pr]. Then the buy/sell points are marked by the points at which this (logarithmic) MACD crosses the zero line, moving up or down respectively.

I would now like to make a slight generalization of the concept of technical indicator, and regard a **technical indicator** as any function of the past prices (and possibly other data), which is supposed to be *correlated* with *future returns*. So, for example, the implication is that the oscillator formed from two moving averages will be above zero when the (intermediate or long-term) future returns are positive, and below zero when they are negative. In other words, there is expected to be a **positive correlation** between this oscillator and the **future returns** over some time interval N. It is possible to form a whole variety of technical indicators of this sort, and measure their correlation with N-day future returns to determine their effectiveness. Then, either a **linear** trading rule can be used in which the position in the security is adjusted to linearly follow the value of the indicator, or a **non-linear** trading rule can be used in which the position is *long* by a fixed amount when the indicator is positive and *short* by a fixed amount when the indicator is negative. (This latter trading rule, of course, requires far fewer trades.) Likewise, the indicator itself can be a **linear** function of the past returns, such as MA's or sums and differences of MA's, or it can be a **non-linear** function of past data, such as polynomials or the hyperbolic tangent function or the error function. By using non-linear functions of the data and measuring their linear correlation with future returns, we are actually capturing some of the **higher-order statistics** of the data, which is probably important for financial data. However, for the time being we will confine the discussion to various linear combinations of various types of smoothings of the past data. However, our method can be extended to non-linear functions of the past data simply by defining and using such functions instead of linear ones. Evidently, some of the traditional technical indicators themselves may be regarded as very complicated non-linear functions of the past price data. Examples of this would be support/resistance levels, head and shoulders tops and bottoms, triangles, rectangles, flags, and so forth, and even trend lines for bull and bear trends [E&M].

I would like to remark here that, in my opinion, most of the traditional rules of **Technical Analysis** are probably obsolete. They probably worked well in decades past, when there were far fewer players in the market and the rate of information exchange was much slower and the amount of information available much less. The markets are undoubtedly much more *efficient* now than they were when these traditional rules were first formulated [E&M]. In particular, the ability to signal a long-term trend change by the crossing of two MA's of much shorter time scale seems "too good to be true", as do the other methods of signaling a trend change by means of technical patterns of short time duration. Probably in today's market the predictive power of any technical indicator formed from price data over a certain time scale is only valid for times of the order of that time scale.

Methods of Smoothing Data

One type of smoothing used in ***QuanTek*** is called the **Savitzky-Golay smoothing** filter. This is a state-of-the-art digital smoothing filter, which has the property that it preserves the first and second moments of the price data. (In other words, if there are peaks in the data, the smoothing preserves the positions of the peaks and also their widths.) This filter uses **Fourier** methods to compute the smoothing. The **Savitzky-Golay smoothing** filter comes in two variants, the **acausal** and **causal** filters. The *acausal* filter smooths over a time window consisting of a number of days *in the past and future* around the given day, equal to the smoothing time period. This *acausal* filter has the advantage that it preserves the phase relationships of the various Fourier components (zero-phase filter). The *causal* smoothing filter smooths over a time window equal to two smoothing periods *in the past*. (This is similar to the usual moving average.) Hence there is an inherent time delay of (approximately) one smoothing period with the causal filter. This causal filter will not preserve phase relationships, which is a disadvantage.

The other type of smoothing used in ***QuanTek*** is **Wavelet smoothing**. This consists of taking a **Multi-Resolution Analysis (MRA)** wavelet transform of the data using the **Wavelet** routines. This is an alternative to the **Fourier** routines used in the **SG** filter, and is necessarily *acausal*. The **Wavelet** method of smoothing is used to construct **technical indicators** or **regressors** that are used in the **Adaptive filter** for the **Price Projection**. Actually, the **Wavelet smoothing** and the **Savitzky-Golay smoothing** are very similar and differ only in details. The **Wavelet** routines are useful in designing an **Adaptive filter** for other reasons, namely that they provide an approximate diagonalization of the **covariance matrix**. In either case, the smoothing has the effect of bringing out the "signal" in the **technical indicators** due to **noise reduction** from the filtering, and increasing the **correlation** between the **technical indicators** and **future returns**.

In addition to **smoothing**, the smoothing filters described above can also be used for **prediction**. The two procedures of smoothing and prediction are closely related. In order to use the filter for **prediction**, the filter must clearly be **causal** (so that it makes a future prediction based only on past data). After de-trending the data, the smoothing filter is applied to produce a smoothed value for future dates relative to the past data set, using the smoothing of the past data. However, using a smoothing filter for prediction makes an implicit assumption about the **correlation** structure of the data, which must be of a very simple type. More sophisticated **Linear Prediction** filters are normally used, which incorporate some means to *estimate* the correlation in the data in order to make the future prediction (see below).

A third type of smoothing is the ordinary exponential **Moving Average (MA)**. This type of smoothing filter is *causal*, in that it does not make use of any data in the future relative to the given day. As is well known, the **exponential MA** also introduces a time delay of the order of one smoothing period (for a time scale of smoothing of two time periods). The **exponential MA** could itself be used to make a future prediction, because technically it is a smoothing filter just like the **Savitzky-Golay** or **Wavelet** smoothing filters. In fact, the exponential MA is actually the optimal **Linear Prediction** filter for the MA(1) process (Moving Average process with an autocorrelation sequence 1 time unit in length) [AH, p.22].

Categories of Smoothed Indicators

There appear to be two basic categories of **technical indicators**, corresponding to two basic categories of **correlation**. The most basic correlation is what is known as **return to the mean**. This implies that there is some *mean* or "correct" price, which the security returns to if the security becomes mis-priced. So, if the price is below some average level, it can be expected to move higher, and if it is above the average level, it can be expected to move lower. So the technical indicator consists of the current price relative to some longer-term average or smoothed price. The future returns are then expected to be *anti-correlated* with this indicator some number of days in the past (or correlated with the *negative* of the indicator). Since the security becomes mis-priced in the first place after some up- or down- move, the presence of a return to the mean mechanism also shows up in the anti-correlation of past *returns* with future returns. There is a rather pronounced **anti-correlation** in daily returns (for some securities) up to about three days in the past, with the future one-day returns, and this can be explained by the return to the mean mechanism acting over these very short time intervals. It also appears to act over much longer time intervals as well. It should be noted that this mechanism is nothing other than the famous "**Buy Low – Sell High**" strategy.

A second correlation is known as **trend persistence**. This correlation corresponds to the tendency of the market to remain in either a **bull** or **bear** market. In other words, if returns are positive or negative in the past, they are the same in the future, so that there is a **positive correlation** between past and future returns. This mechanism would seem to be at variance with the return to the mean mechanism, which implies **negative correlation**. However, these two mechanisms can be reconciled by supposing that the "mean" is some smooth, *slowly* varying function of past prices and economic data. The **trend**, corresponding to a bull or bear market, is **persistent** and is related to the (usually) slowly varying rate of change of this price mean. (Or it can be thought of as the mean value of the returns.) Then, the shorter-term **fluctuations** about this mean price level are **anti-persistent**, and correspond to the **return-to-the-mean** mechanism. So, given any time scale, we may smooth the price data on this time scale, and then suppose that the smoothed long-term trend is persistent, and the short-term fluctuations about this trend are anti-persistent. Evidently in an efficient market, these two mechanisms "cancel out" on all time scales, leading to zero correlation and neither persistence nor anti-persistence of returns. But when inefficiencies exist, they do not cancel out, and correlation may exist on certain time scales. Evidently the true situation is much more complicated than this, and what has just been said should be regarded as merely an oversimplified "sketch" of the true picture. To our knowledge, nobody has yet formulated a complete theory of stock price correlations, although steps in this direction are outlined in <u>The Econometrics of Financial Markets</u> (1997) by Cambell, Lo, & MacKinlay [CLM].

A possible third correlation does not really have a name, but we will call it the presence of **turning points** or **trend reversal** mechanism. According to this idea, if we can identify the turning points or changes of trend of the price data, then this will be correlated with a future positive or negative trend. In other words, if we can identify a point where the trend seems to change from negative to positive, then this should be correlated with a future positive trend, and a point where the trend seems to change from positive to negative should be correlated with a future negative trend. Examples of these change-of-trend indicators in traditional **Technical Analysis** are identification of top and bottom formations such as **head-and-shoulders**. However, we may also construct an oscillator-type indicator by taking the rate-of-change of the returns, which is itself a rate-of-change of the log prices. In other words, the returns are the **first derivative (velocity)** of the log prices, while the rate-of-change of returns is the **second derivative (acceleration)** of the log prices. The hypothesis is then that this **turning point indicator** is correlated with future returns, at some point in the future. However, this indicator may be less reliable than the first two, because it tends to emphasize the higher frequency modes, while most of the correlation seems to exist in the low frequency modes.

Phase of Smoothed Indicators

When constructing the above described indicators using **acausal** smoothing, so the **phase** of the indicator coincides with the data (zero-phase filter), one would expect a definite phase relationship between the indicator and future returns. However, it is necessary now to discuss a subtle point. It is the distinction between *deterministic* and *random* signals. It is a mathematical theorem by Kolmogorov [B&D] that if a signal has a spectrum that is zero in any finite region (on a region of "non-zero measure") then the signal must be *deterministic*. In particular, this will be the case if the spectrum is *discrete*. On the other hand, if the signal has a continuous spectrum that is greater than zero on all but a "set of measure zero" (i.e., a set of discrete points) then the signal must be *random*. What does this have to do with stock trading? In both the **SG smoothing** and the **Wavelet smoothing**, the signal is treated as a "circular" set of discrete data of length, say, 2048. These smoothing routines then separate the signal into a *discrete* set of frequencies. If this is actually possible for the real spectrum, then the signal is necessarily *deterministic*. This means that it has to be *perfectly predictable*. We know this is not the case, of course, with real returns data. If it were perfectly predictable, then we could just separate out one particular frequency, trade in phase with the smoothed indicator of that one frequency, and make a killing! The other frequencies would cancel out, and the trading cycle with the one frequency we select would lead to huge gains!

Why is this not possible? The **Random Walk** model says that the returns data are completely *random* and have no predictive power. When we separate the returns data into *discrete* frequencies, evidently this is only an approximation, as random data must have a *continuous* spectrum. So the spectrum actually consists of a continuum of frequencies, with amplitudes and phases that vary, either continuously or discontinuously. In either case, evidently the *phase* of the discrete frequency components of the **SG** or **Wavelet** transform is what varies randomly. In that case, trading using this single frequency component with a random phase, cannot lead to any gains, consistent with the **Random Walk** model.

Are the signals *deterministic* or *random*? The **Random Walk** model says they are random, but it could be that there is also a deterministic component superposed, at least for short periods of time and certain frequencies. Then the **buy/sell points** determined by the **SG** or **Wavelet** transform of the returns and resulting **technical indicators** might have some predictive power, if you happen to choose the correct trading time scale at the correct time. At any rate, these buy/sell signals should not be any *worse* than random timing, provided the main criterion for trading is the expected future return or portfolio rebalancing. At worst, the buy/sell signals are not predictive and any buy/sell points are as good as any other, as long as the decision to buy or sell is made according to some other criterion. But in some cases, as **Technical Analysts** believe, it might be possible to achieve gains by "timing" the market. If the **buy/sell signals** are used as timing points to *rebalance* the portfolio, as opposed to, say, doing it every month, then this should do no harm and might do some good.

1.5 The Harmonic Oscillator Indicator

There are three main types of **technical indicators** used in ***QuanTek***. These correspond to the three main types of correlation mentioned above, namely **return to the mean** and **trend persistence**, plus **turning point** or **trend reversal**. These three types of indicators are implemented by means of the **Savitzky-Golay** or **Wavelet smoothing** filter, using the filter directly on the (logarithmic) price data to obtain the **Relative Price** indicator, taking the first derivative to obtain the **Velocity** indicator, or taking the second derivative to obtain the **Acceleration** indicator. These indicators are also projected ahead in time using a **Standard Linear Prediction (LP) filter**. They are incorporated in the **Harmonic Oscillator** indicators displayed in a splitter window, and are used mainly to identify possible **buy/sell points**. The **Wavelet smoothing**, on the other hand, is also used to construct these indicators to be used as **regressors** in the **Adaptive filter**, which are constructed using circular boundary conditions.

In the description of the indicators given below, the *phase* relationships between the **Relative Price**, **Velocity**, and **Acceleration** indicators with respect to the **returns** are given in a theoretical way. These descriptions would be accurate if the returns followed a pure sine wave of a single frequency, and the **time horizon** and **smoothing time scale** were set according to this frequency (the period of the wave equal to $2N$ days). These are the phase relationships *assumed* in computing the **buy/sell points** from the **Harmonic Oscillator** indicators. However, when the correlation of these three indicators with the **future returns** is computed, it is found that these phase relationships often do not hold exactly. The indicators have a different phase relationship to the *N*-day future returns than they theoretically should. The origin of this phase discrepancy no doubt has to do with the fact that the spectrum of the security returns consists of a continuous spectrum of waves of all different frequencies, not just of one frequency, and the **amplitude-phase relationships** between these different waves can be complex and rapidly varying. (In fact, the **randomness** of the data can be re-interpreted in these terms. If these **amplitude-phase relationships** are stable for a period of time, then this represents **correlation** in the data.) To take this into account in the **Adaptive filter**, an adaptive fit is made to the **Relative Price** and **Velocity** indicators, and perhaps also the **Acceleration** indicator, which are 90 degrees out of phase with each other (theoretically). Time-varying sums of these indicators in the fit can thus take into account the (time-varying) phase relationships between the **indicators** and **future returns**, and if the relationships are **persistent**, then this should lead to **predictive power**.

Relative Price Indicator

The first type of indicator that might be used to form a **technical indicator** is called the **Relative Price**. This is a difference of the (logarithmic) price levels with smoothings on two different time scales, the shorter time scale minus the longer time scale. This is similar to the **MACD** oscillator consisting of the difference of two exponential MAs mentioned previously (except without the time lag). This type of indicator is a measure of the **return to the mean** mechanism, with the longer time period smoothed price level playing the role of the mean level. When the shorter time period smoothed price level is below the longer period one, the future prices are expected to *rise*, and when it is above, the future prices are expected to *fall*. There is a certain time delay here, which is of the order of the shorter smoothing time period, in which the trough or peak of this indicator *now* implies that the future returns will be positive or negative *later*, roughly by this time delay, which corresponds to one *time period* of $N/2$ days, with N being the shorter **smoothing time scale**. So there is a *phase difference* between the **Relative Price** indicator and the future returns that it is supposed to predict. The negative of this indicator *leads* the expected future returns by *approximately* one *time period* of $N/2$ days (or 90 degrees), so if the trading time scale is N days, this indicator will be correlated with the *future* N-day returns.

Due to this phase relationship, we expect the **buy points** to pass through the *minima* [**min**] of the **Relative Price** indicator, and the **sell points** to pass through the *maxima* [**max**] of this indicator, for an N-day **time horizon**. This is, of course, nothing other than the **Buy Low – Sell High** mechanism at work. In ***QuanTek***, a **Relative Price** indicator is displayed as part of the **Harmonic Oscillator** indicator in a splitter window. The **buy/sell points** that are displayed as vertical green/red lines are *defined* by the minima/ maxima of the **Relative Price** indicator (along with the rest of the **Harmonic Oscillator**). Note that, unlike the exponential moving average, the **Savitzky-Golay** or **Wavelet smoothing** filter used in this indicator is *acausal* and has no time lag. The downside of this is that it requires a future **Price Projection**, for which we use the **Standard LP filter**, and the smoothing mixes some of the future projection with the past data (so it is not *causal*).

Velocity Indicator

The second type of indicator that might be used to form a **technical indicator** is called the **Velocity**. It is the smoothed **first derivative** of the log prices. This indicator is a measure of **trend persistence**. It is clear that if the trend is persistent, then the smoothed **Velocity** of the log prices should be correlated with the **future returns**. So the **Velocity** indicator is *in phase* with the **future returns**.

The **buy points** will thus correspond to the points where the returns start to become positive, in other words the zero-crossing points from negative to positive [**Z+**]. Likewise, the **sell points** will correspond to the points where the returns start to become negative, which are the zero-crossing points from positive to negative [**Z**–]. Since the *negative* of the **Relative Price** indicator *leads* the returns, the trough of the **Relative Price** [**min**] corresponds to the positive zero-crossing point [**Z+**] of the **Velocity** indicator, which is a **buy point**. Likewise, the peak of the **Relative Price** [**max**] corresponds to the negative zero-crossing point [**Z**–] of the **Velocity** indicator, which is a **sell point**. So the green/red vertical lines denoting **buy/sell points** pass through these points of the **Relative Price** and **Velocity** indicators. The **Velocity** indicator, since it is supposed to be correlated with **present returns**, should be *in phase* with the N-day returns. Hence the **Velocity** indicator will *lag* by *approximately* one *time period* ($N/2$ days, or 90 degrees) the start of the **future *N*-day returns**. In other words, the **Velocity** indicator will reach a peak in the middle of the *N*-day period of positive returns.

Acceleration Indicator

We may also construct a third type of smoothing that might be used to form a **technical indicator**, which is called the **Acceleration**. It is the smoothed **second derivative** of the log prices. This indicator may be interpreted as an indicator of **turning points**, because the second derivative is positive when the prices are at a minimum (positive or upward curvature) and is negative when they are at a maximum (negative or downward curvature). Hence it can be seen that this **Acceleration** indicator will be *positive* when the **Relative Price** is *negative*, and vice-versa. Thus the **Acceleration** indicator is exactly *out of phase* with the **Relative Price**. However, the **Acceleration** differs from the **Relative Price** in that, with each successive derivative, the high frequency components are emphasized more and more. Hence the **Acceleration** indicator contains much more of the high-frequency components than the **Relative Price** and hence is much less smooth. The **Acceleration**, since it indicates a *turning point*, should *lead* the **returns** by approximately one *time period* ($N/2$ days or 90 degrees). A positive **Acceleration** peak, indicating a turning point from negative to positive returns, should be followed approximately $N/2$ days later by a positive peak in the **Velocity**, and likewise a negative **Acceleration** trough, indicating a turning point from positive to negative returns, should be followed approximately $N/2$ days later by a negative trough in the **Velocity**. Hence the **buy points** will correspond to positive peaks [**max**] in the **Acceleration**, and the **sell points** will correspond to negative troughs [**min**]. So the **Acceleration** is 180 degrees out of phase with the **Relative Price**, and *leads* the **Velocity** by 90 degrees or one *time period*. The green and red vertical lines denoting the **buy/sell points** indeed pass, approximately, through the peaks/troughs of the **Acceleration** indicator. The peak of the **Acceleration** indicator should hence occur approximately at the start of an *N*-day period of positive returns.

1.6 Price Projections & Linear Prediction Filters

There are many different varieties of **Linear Prediction (LP) Filters** available for use. The simplest types presume that the data satisfy **stationary statistics**, meaning that the statistical properties of the data are assumed constant in time. This condition is satisfied in many different systems, for example in studies of the 11-year sunspot cycle, but unfortunately not for financial data. So although these simple stationary LP filters may be used for **Linear Prediction** of financial data, the results will usually not be very good (although in some circumstances they do work well). Most of the time, the financial data satisfy **non-stationary statistics**. This means that the **mean** and **covariance matrix** of the **returns** changes with time. In order to track these non-stationary statistical properties, an **Adaptive filter** such as the **Least-Mean Square (LMS) filter** may be used.

Another problem with financial data is that the data are almost entirely noise. The problem is to ferret out any small correlation in the returns, which are then used to estimate the future returns. If the data really are white noise – with no correlation – then the **Random Walk** model holds and the future projection is just the average **drift** or **mean** of the returns. The usual methods of estimating the covariance matrix yield a fit to the noise rather than the actual correlation. The way to reduce the noise is to use smoothing to create smoothed **indicators**, and then measure the (time-varying) correlation between these indicators and the future returns. However, in order to understand these noise-reduction methods, it is helpful to first understand the basic LP filters based on **stationary statistics** and the usual definition of the covariance matrix. Thus we begin with the basic **Yule-Walker equations** for estimating the future return in terms of past returns. More generally, the **Wiener-Hopf equations** are used when an independent estimate of the **covariance matrix** is available. Note that these filters only make use of **stationary second-order statistics**, namely the **mean** and **covariance** of the returns, which are assumed **stationary**. A more general **Adaptive Filter**, utilizing **technical indicators** which can be arbitrary smoothed functions of the past data, can utilize **higher-order statistics** that are **non-stationary**. For example, one of the indicators could be the squared returns, or **volatility (variance)**, and then the filter would utilize the **covariance** between this indictor and the future returns. This would then incorporate a type of **GARCH** (**Generalized Auto-Regressive Conditional Heteroskedasticity)** model. However, we begin with the most basic type of **LP filter**, utilizing only the covariance between the returns themselves.

Covariance Matrix & Yule-Walker Equations

Given a **stationary stochastic process**, the **Linear Prediction** filter is derived making use of the **second-order correlation** in the (logarithmic returns) data. This correlation is assumed constant in time, due to stationarity. On a theoretical level, the time series is modeled as an **auto-regressive** time series which obeys the **Yule-Walker equations**. To derive these equations, we need to pretend as if there is an infinite **statistical ensemble** of realizations of a given stochastic process, or in other words, the **stochastic process** specifies the statistical properties of the time series, and we can generate an unlimited number of actual time series with these statistical properties, starting with a different set of random numbers as input for each time series. Then we can define, theoretically, an **expectation value** over this statistical ensemble of realizations of the stochastic process, as an average of some given quantity over the whole ensemble. In practice, these expectation values may be approximated by various sums over the available data set, such as for example the **sample mean** and **sample covariance**.

Now, using the notation of Haykin [Hay], suppose we are given a financial time series of returns of length *N*. The *N*-by-1 *observation vector* is defined by [Hay]:

$$\mathbf{u}(n) \equiv \left[u(n), u(n-1), \ldots, u(n-N+1)\right]^T$$

In this time series, the latest date is denoted by the index *n*, and the index increases moving forward in time. The observation vector is labeled by its latest date *n*.

We first assume that the time series is **stationary**, so the covariance matrix of the returns is independent of time (hence independent of the index *n*). The (stationary) **covariance matrix** is then defined by the following expectation value with respect to the statistical ensemble:

$$\mathbf{R} \equiv E\left[\mathbf{u}(n)\mathbf{u}^T(n)\right]$$

If the covariance matrix is **stationary**, then it depends only on the *difference* of the indices of the components of the two observation vectors. Correspondingly the **autocovariance function** or **autocovariance sequence** is defined as the expectation:

$$r(k) \equiv E\left[u(n)u(n-k)\right]$$

Then in terms of the autocovariance sequence *r*(*k*), the **covariance matrix** is given by:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & r(N-1) \\ r(1) & r(0) & \cdots & r(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(N-1) & r(N-2) & \cdots & r(0) \end{bmatrix}$$

Note again that due to stationarity, the coefficients *r*(*k*) of the autocovariance sequence or matrix depend only on the *time lag* between the two elements $u(n)$ and $u(n-k)$ of the time series, not on the index *n* of the series. This is fortunate, for otherwise we would not be able to approximate these quantities by the corresponding *sample expectation values*. The **sample autocovariance sequence** is then given by the following sum over the time series values, separated by the time lag *k*:

$$\hat{r}(k,N) \equiv \frac{1}{N}\sum_{n=0}^{N-1} u(n)u(n-k)$$

Of course, for a data set of length *N*, this sum runs out of the bounds of the data set. So the above definition should be interpreted as applying to an infinitely long time series, from which we extract a sum over *N* terms. For a data set of finite length, this definition can be modified appropriately.

Now we suppose that the future value of the time series can be (partially) predicted as a linear sum over past values of the time series, plus a random *white noise* term. (The **Random Walk** model corresponds to the *white noise* term alone.) A stochastic model, in which the future value of a variable in a time series is determined as a linear function of *M* past values, plus additive white noise, is called an **autoregressive process (AR)** of order *M*. This stochastic process satisfies the following difference equation [Hay]:

$$u(n) = \nu(n) + w_1 u(n-1) + w_2 u(n-2) + \cdots + w_M u(n-M)$$

The additive *white noise* $\nu(n)$ is assumed to be a Gaussian random variable of **zero mean** and **constant variance**, uncorrelated for different times:

$$E\left[\nu(n_1)\nu(n_2)\right] = \begin{cases} \sigma_\nu^2, & n_1 = n_2 \\ 0, & \text{otherwise} \end{cases}$$

Here, σ_ν^2 is the **noise variance**.

Next we take the expectation value of the autoregressive process with $u(n-k)$ on both sides of the equation. Making use of the fact that the white noise term $\nu(n)$ is uncorrelated with anything with an index different from *n*, we arrive at the following result:

$$E\left[u(n),u(n-k)\right] = w_1 E\left[u(n-1),u(n-k)\right] + \cdots + w_M E\left[u(n-M),u(n-k)\right]$$

Using the definition of the autocovariance sequence, we may then rewrite this in terms of the autocovariance sequence itself:

$$r(k) = w_1 r(k-1) + w_2 r(k-2) + \cdots + w_M r(k-M)$$

This may then be written in explicit matrix form as follows, noting that due to stationarity we have $r(k) \equiv r(-k)$:

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r(1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(M-1) & r(M-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(M) \end{bmatrix}$$

This set of equations is called the **Yule-Walker equations**. They may be expressed for simplicity in explicit matrix form. Then, assuming the covariance matrix is *nonsingular*, it may be inverted and we may solve for the **Linear Prediction (LP) coefficients** as follows:

$$\mathbf{Rw} = \mathbf{r} \qquad \Rightarrow \qquad \mathbf{w} = \mathbf{R}^{-1}\mathbf{r}$$

Hence the coefficients of the stationary autoregressive process may be obtained from the covariance matrix, provided that it is not singular, by a simple matrix inversion. These are the basic equations for the **Linear Prediction Filter**. In the case of the **Yule-Walker equations**, the **stationary covariance matrix** is estimated as a sum over products of past values of the time series, with a definite form. It should be noted from the above definition of the covariance matrix that the matrix is *symmetric* and all the elements on each (major) diagonal are equal. This type of matrix is called a **Toeplitz** matrix.

The matrix inversion process described above for the computation of the LP coefficients can be problematic, in general, for a large matrix. But for a **Toeplitz** matrix there exist fast routines that can be used for their inversion numerically. Thus, once the autocovariance sequence is estimated, and from it the covariance matrix is formed, the covariance matrix can be inverted using this routine to obtain the LP coefficients from the product of the inverted covariance matrix and the autocovariance sequence.

Estimating the Covariance Matrix

In the general case, we are faced with the problem of finding some estimate the covariance matrix. In order to estimate the (stationary) covariance matrix, there are at least three methods that can be utilized. Probably the simplest method is to just compute the sample autocovariance using the formula given above. More generally, the sample **mean** and **covariance** are given by the following formulas, for a data sequence $X_0 \ldots X_{N-1}$, for a (finite) sample size of *N*. These yield a non-negative definite covariance matrix [B&D]:

$$\bar{X} \equiv \frac{1}{n}\sum_{t=1}^{N} X_t \qquad 0 \le k \le N-1$$
$$\hat{\gamma}(k) \equiv \frac{1}{N}\sum_{t=0}^{(N-1)-k} \left(X_t - \bar{X}\right)\left(X_{t+k} - \bar{X}\right)$$

Note that the element $\hat{\gamma}(0)$ of the covariance sequence given above is called the **variance**. This is a measure of the total noise power in the time series. In practice, even though the autocovariance sequence $\hat{\gamma}(k)$ defined above is defined for all values of k such that $0 \le k \le N-1$, in practice only the first half of the values are used, because it can be seen that the second half of the values use an increasingly smaller number of the time series elements in their definition. In fact, the last element of the covariance sequence is just the product of the first and last elements of the time series (minus the mean), according to the above formula.

An alternative method for estimating the **covariance matrix** is by means of the **spectrum**. The **spectrum** can be estimated either using the **Fast Fourier Transform (FFT)** or the **Discrete Wavelet Transform (DWT)**. (Also the **Maximal Overlap Discrete Wavelet Transform (MODWT)** can be used.) The **FFT** is a transformation in which the data set of daily returns, as a function of time, is decomposed into its component frequencies in an interval of fixed length, which we take to be 2048 days for the filter routines. The output of the **FFT** routine, for a date length of 2048 days of returns, yields an *amplitude* for each frequency interval between 0 frequency and a frequency of 1 cycle per 2 days (Nyquist frequency), which is the maximum frequency possible for daily time series. There are 1024 evenly spaced frequency intervals in this range for a data set of 2048 days. The other half of the output consists of 1024 values of the *phase*, which are not used. The spectrum is then obtained by squaring the amplitude, which yields the spectral power at each frequency. The **FFT** thus represents the time series as a sum of sine waves of constant amplitude over the 2048-day range. The **DWT**, on the other hand, works a little differently. In this case, the data set of 2048 days of returns data is decomposed in the **wavelet** basis, which is a set of waves, which are localized in both frequency and time, but of finite extent in both. In this **wavelet** basis, the returns data are represented as amplitudes as a function of *time*, decomposed into a set of 8 *frequency octaves*. These amplitudes are then squared to get the spectral power corresponding to each wavelet component.

The point of computing the **spectrum** is to detect if there is any **correlation** present. If there is no **correlation**, and the time series is just random **white noise**, then the *true* power spectrum will be flat or constant. However, the *measured* power spectrum, from the **FFT** or **DWT**, is not generally constant due to *stochastic noise*. In fact, it is shown in standard textbooks on time series [B&D] that for the case of the **FFT**, the variance of each value of the power spectrum, for each discrete frequency, is 100%, or in other words, totally uncertain. In order to uncover the true spectrum, therefore, the **FFT** spectrum must be *smoothed*, or averaged over a number of frequency values. This connection between a variation of the **spectrum** and the presence of **correlation** in the time series is made explicit by the **Wiener-Khinchin Theorem**, a nice (short) description of which given in *Numerical Recipes* [NR]. This theorem simply states that the **power spectrum** is the **Fourier Transform** of the **autocovariance function**, and vice-versa. Then since the **covariance matrix** is built from the **autocovariance function** as described earlier, by taking the **Fourier Transform** of the power spectrum, we arrive at the autocovariance function and hence the covariance matrix. However, it must be emphasized that these spectral methods of estimating the covariance matrix only work if the time series is *stationary*.

For a **non-stationary** time series, we can break the time series up into segments of length N and approximate the covariance matrix in each segment as **stationary**. The length N of the segments must be chosen to be short enough so that the covariance matrix is approximately stationary in each segment, but not so short that the covariance matrix is dominated by stochastic noise. In the **Least-Mean Square (LMS) adaptive filter** described below, the covariance matrix is actually approximated by a segment consisting of a single term for each time step, so it is very noisy. But the noise cancels out in the adaptation of the filter over time. Alternatively, the covariance matrix could be approximated by a sum over a short segment of the data in this **adaptive filter** routine, to help reduce the **stochastic noise**. We also can reduce the **stochastic noise** by computing the covariance between various **smoothed technical indicators** and the ***M*-day returns**, rather than the covariance between individual terms in the returns sequence as was done above.

Wiener-Hopf Linear Prediction (LP) Filters

From the assumed **auto-regressive** model of the time series of returns as given by the **Yule-Walker** equations, we may develop a **Linear Prediction filter** to estimate future returns in the time series. Let us now change the notation somewhat, and let the index n denote the *present* time, with the index *increasing* in the *past* and *decreasing* in the *future*. Hence for the case $n = 0$, past indexes are denoted by positive values and future indexes by negative values. We now wish to form an *estimate* of the *M*-day future returns, at the date *n*, which we denote by $\hat{y}_{(n)}^{M}$, as a regression on the past returns. The LP coefficients are denoted by $w_k \left(k \geq 0\right)$. The regression is performed over the past *N* days of returns. Thus we have the equation:

$$\hat{y}_{(n)}^{M} = \sum_{k=0}^{N-1} w_k^{(n)} u_{n+k}$$

The task is to calculate the optimal LP coefficients $w_k^{(n)}$ at time *n*. To do this we must compare the *M*-day future projection with the "actual" future *M*-day returns. (Obviously this comparison is made over past data.) We denote the "desired response" or actual future *M*-day return with respect to (present) time *n* by $d_{(n)}^{M}$:

$$d_{(n)}^{M} \equiv \sum_{m=1}^{M} u_{n-m}$$

Now we may define an estimation error:

$$e_{(n)}^{M} \equiv d_{(n)}^{M} - \hat{y}_{(n)}^{M}$$

In the **Wiener-Hopf** theory, we choose to minimize the mean-square value of this error. Thus we minimize a *cost function* defined by [Hay, p.97]:

$$J_{(n)}^{M} \equiv E\left[\left|e_{(n)}^{M}\right|^{2}\right]$$

Thus the LP coefficients are adjusted so that the cost function is minimized, for each date *n*. This gives the optimal estimate of the future returns with respect to time *n*, in the least-square sense.

The optimal filter coefficients are found by varying the cost function with respect to the filter coefficients, and setting this variation to zero. The cost function may be expressed in terms of the above equations as:

$$J_{(n)}^{M} \equiv E\left[\left|d_{(n)}^{M} - \sum_{k=0}^{N-1} w_k^{(n)} u_{n+k}\right|^{2}\right]$$

Thus the variation with respect to the LP coefficients yields:

$$\frac{\partial J_{(n)}^{M}}{\partial w_j^{(n)}} \equiv E\left[-2\left(d_{(n)}^{M} - \sum_{k=0}^{N-1} w_k^{(n)} u_{n+k}\right) u_{n+j}\right] \equiv 0$$

This may be rewritten in the following form:

$$E\left[d_{(n)}^{M}u_{n+j}\right]-\sum_{k=0}^{N-1}w_{k}^{(n)}E\left[u_{n+k}u_{n+j}\right]\equiv 0$$

These equations are an expression of the *Principle of Orthogonality* [Hay, p.99], which states that for the optimal **Wiener-Hopf filter** the estimation error is orthogonal (uncorrelated) to all the input samples, $E\left[e_{(n)}^{M}u_{n+j}\right]\equiv 0$.

Let us now define some new notation. The covariance matrix of the past returns and the covariance between the past returns and future *M*-day returns at date *n* is given by:

$$\Gamma_{k,j}^{(n)}\equiv E\left[u_{n+k}u_{n+j}\right]\qquad \hat{\Gamma}_{j}^{(n)}\equiv E\left[d_{(n)}^{M}u_{n+j}\right]$$

In general, these expectation values may be estimated by a variety of methods as discussed in the previous section. They are not restricted to the particular form that was given for the **Yule-Walker** equations. In general, the covariance matrix is always symmetric, assumed non-singular, but need not be **Toeplitz** as in the case of the **Yule-Walker** equations.

Thus the above equation may be written in terms of these covariance matrices as:

$$\hat{\Gamma}_{j}^{(n)}-\sum_{k=0}^{N-1}w_{k}^{(n)}\Gamma_{k,j}^{(n)}\equiv 0$$

The hat is written over the future covariance matrix because, in general, it must be estimated or extrapolated from past data. Thus for the present time $n=0$, and taking into account that the past covariance matrix is symmetric, we may rewrite this as:

$$\sum_{k=0}^{N-1}\Gamma_{j,k}w_{k}^{(0)}=\hat{\Gamma}_{j}$$

Now we assume the covariance matrix is nonsingular, so it can be inverted (as should always be the case if the problem is well-posed). Doing this, we arrive at the **Wiener-Hopf** equations:

$$\boxed{w_{k}^{(0)}=\sum_{j=0}^{N-1}\Gamma_{k,j}^{-1}\hat{\Gamma}_{j}}$$

This represents the solution to the optimal filter, but note that the future covariance matrix $\hat{\Gamma}_{j}$ must be estimated somehow, since the covariance between past and future returns is unknown at the present date $n=0$. Normally this is extrapolated from its past values, using an algorithm such as an **Adaptive filter**. (If the time series is **stationary**, then it is *equal* to its past values.)

Least-Mean-Square (LMS) Adaptive Filters

If the time series data are **non-stationary**, then we need to estimate the LP filter coefficients recursively, starting at some past time and allowing the coefficients to "adapt" to the changing environment by means of the recursive procedure. Thus the **Least Mean Square (LMS) filter** involves a kind of "training" procedure over the history of the returns data. This is done by defining the cost function given above and then "adapting" the LP coefficients to move toward the minimum of the cost function recursively. This may also be viewed as a method of computing the covariance matrices given above.

The gradient of the cost function with respect to the LP coefficients, as given above, is:

$$\frac{\partial J_{(n)}^{M}}{\partial w_{j}^{(n)}} \equiv -2E\left[\left(d_{(n)}^{M} - \sum_{k=0}^{N-1} w_{k}^{(n)} u_{n+k}\right) u_{n+j}\right]$$

In the *Method of Steepest Descent* [Hay, p.206], the LP coefficient vector $w_j^{(n)}$ is adjusted in successive steps, to try to reach a minimum of the cost function $J_{(n)}^{M}$. This is done by adjusting $w_j^{(n)}$ in the downhill direction of $J_{(n)}^{M}$, which is the direction of the negative gradient of $J_{(n)}^{M}$. The *instantaneous estimate* of $J_{(n)}^{M}$ is used, rather than the expectation value as given above, so the expectation value symbol is dropped and we define the instantaneous estimate as:

$$-\frac{1}{2}\frac{\partial \hat{J}_{(n)}^{M}}{\partial w_{j}^{(n)}} \equiv \left(d_{(n)}^{M} - \sum_{k=0}^{N-1} w_{k}^{(n)} u_{n+k}\right) u_{n+j}$$

Now, given that we have defined time to move forward in the negative direction, we define the recursion for each time step according to:

$$w_{j}^{(n-1)} \equiv w_{j}^{(n)} - \frac{\mu}{2}\frac{\partial \hat{J}_{(n)}^{M}}{\partial w_{j}^{(n)}} = w_{j}^{(n)} + \mu u_{n+j}\left(d_{(n)}^{M} - \sum_{k=0}^{N-1} u_{n+k} w_{k}^{(n)}\right)$$

The parameter μ determines the size of the step at each recursion, and has a crucial effect on the performance of the adaptive filter [Hay]. This is the basic recursion for the **Least Mean Square (LMS) filter**. It involves the instantaneous values for the covariance matrices defined previously, rather than the expressions in terms of expectation values [Hay, p.236]. Notice that the term in parentheses is just the estimation error at date *n*. The rate of adaptation of the LP coefficients is thus proportional to the estimation error. This rate is also controlled by the parameter μ, which must be set correctly [Hay].

In the above algorithm, the future 1-day return is regressed on the entire set of past 1-day returns. However, for noisy financial data, this may result in too much “fitting to the noise”. It makes sense to modify the above algorithm to regress on only a small number of smoothed functions of the past data, rather than the entire set of past returns. Also we estimate the *M*-day returns rather than a sequence of *M* 1-day returns. This will then result in noise reduction and a better fit to the “true” signal. So instead of regressing on the past returns u_n, we take a small set of “regressors” or **technical indicators** as the regression variables, which are functions of the past returns (and possibly other data). The indexes of the LP coefficients then run only over this small set of regressors, not over the entire set of past daily returns. The past covariance matrix is that of the regressors, and the future covariance matrix is the vector of covariances between the regressors and the future *M*-day return. The adaptive algorithm still runs in 1-day steps over the past daily data. Other than that, the problem is the same. The reduction in regression variables simplifies the estimation of the past and future covariance matrix. The problem of estimating the covariances is also simplified by using the **Wavelet** decomposition, since the parts of the **technical indicators** on different wavelet levels can be taken to be orthogonal.

1.7 Statistical Tests – LP Filters & Indicators

QuanTek has a variety of statistical tests and displays, designed to measure quantities of interest in **Econometrics** and **Time Series Analysis**. The first two tests are designed to directly detect the presence of **correlation** between functions of the past returns, and the future returns. The first dialog box is the **Indicators – Linear Prediction Filters** dialog, on the **Statistical Tests** dialog (**Correlation – LP Filter** button), and the second is the **Correlation Test** – **LP Filters** & **Indicators** dialog. This second dialog is reached by clicking the **Correlation** button on the first dialog. To go back to the first dialog, click the **Close** button in the second dialog. These two dialogs work in tandem with each other. In the **Indicators – Linear Prediction Filters** dialog, you select one of six **technical indicators**, and then apply various degrees of smoothing and filtering to them. Then by going to the **Correlation Test** – **LP Filters** & **Indicators** dialog, you can measure the degree of **correlation** between these indicators and future returns. The method of measuring this correlation is described in Method of Correlation Tests (below).

Indicators – Linear Prediction Filters

After selecting a stock data file and **Adaptive filter** in the **Choose LP Filter** dialog on the **Statistical Tests** dialog, you can display one of six **technical indicators** in the **Indicators – Linear Prediction Filters** dialog. The first three indicators are the raw output of the **Adaptive filter**, the raw output of the **Standard LP filter**, and the simple **Long-Term Trend** line. These indicators are not smoothed or filtered. The second three indicators are the **Relative Price**, **Velocity (Returns)**, and **Acceleration** indicators. To these you can apply **Wavelet Smoothing** on various time scales, both **Low-Pass** and **Band-Pass**. There is a graph for viewing these indicators over the past 1024 days. The graph is in the form of a set of past values out to –400 days plus future values, out to +100 days. This entire indicator consisting of past and future values can be recalculated for any day in the past and displayed in the graph. This is the purpose of the **Indicator in the historical past** spin control. Note that for the second three indicators, the future values are provided by the **Standard LP filter**, which is the standard Burg Order-Regressive LP filter available commercially [NR]. After selecting the technical indicator and smoothing, you can click the **Correlation** button to call up the **Correlation Test** – **LP Filters** & **Indicators** dialog. This dialog then computes the **correlation** between the chosen **indicator** and **future returns** for the given security. In particular, choosing the **Adaptive filter**, you can measure the effectiveness of this filter directly in terms of the degree of **correlation** between its prediction and the actual future data.

Correlation Test – LP Filters & Indicators

After defining a technical indicator in the **Indicators – Linear Prediction Filters** dialog, you can click the **Correlation** button to call up the **Correlation Test – LP Filters & Indicators** dialog, which displays the **correlation** between the **technical indicator** and the **future returns**, as a function of the **time lag**. How this works is described in the next section, Method of Correlation Tests. The indicator consists of a range of past and future values, corresponding to either the actual past returns with a future projection added on, or various types of smoothing and filtering of these. For each value, the **correlation** with the ***N*-day return** is measured, where ***N*** is the value set in the **Time Horizon** control. This yields a graph that, hopefully, displays a nice peak centered on the vertical line labeled ZERO (the present, no time lag). This graph may be moved left or right by means of the **Lead Time** control, to display different parts of the graph.

The correlation is measured over a set of past data whose length is set in the **Correlation Scale** list box. The correlation can also be measured at any past time using the **Correlation in historical past** spin control. This is extremely useful for seeing graphically how the correlation changes over time (non-stationarity). This is also a good way to gauge the **persistence** of the correlation, to determine whether the correlation is real or merely stochastic noise. (It is particularly interesting to compare the **persistence** of the **correlation** of the **Adaptive filter** output with future returns, versus the correlation of the simple **Long-Term Trend** line with future returns. Remember that the long-term investor is basically using the **Long-Term Trend** line as their **Price Projection**.)

The **Correlation Test – LP Filters & Indicators** dialog box also computes some numerical quantities of interest. First, the actual value of the **correlation** under the ZERO line is displayed. The **standard error** is displayed, which depends on the number of data points. The average ***N*-day volatility** for the stock data returns is displayed. From these numbers, a theoretical estimate of the annual simple and compound gain is computed, using the formula: daily correlation times daily volatility times 256 for simple returns. This gives you an idea of the theoretical gains possible with a given degree of correlation, given the volatility of the returns for that stock. There is also a set of radio buttons to change the vertical scale of the graph.

Method of Correlation Tests

The correlation displays in the **Correlation Test – LP Filters & Indicators** test are of an unusual type. They give the correlation as a function of the *time lag* of the indicator or filter output. It is important to explain what this means. Let us consider first the output of the **Linear Prediction** filter, and explain how the data in ***QuanTek*** is arranged. The past logarithmic price data are labeled by an index which is 0 for the *present* day, or most recent day of daily data. Then the index becomes more positive going backward in time, and more negative going forward in time. The index labeling the *future* **Price Projection**, on the other hand, is negative. The indexes for the past and future price data may be illustrated in the following diagram:

…	**+6**	**+5**	**+4**	**+3**	**+2**	**+1**	**0**	**–1**	**–2**	**–3**	**…**

The past data indexes are shown in black, while the future data indexes are shown in blue. This is analogous to the Price Graph in ***QuanTek***, where (with the white background) the past data are shown in black and the future projection is in blue. The *returns* are the changes in log price from one (close) log price to the next day's (close) log price. The returns are also labeled by the above indexes, with the index corresponding to the later day of the price difference. (The return with index 0 is the log price for day 0 minus the log price for day +1.)

Now, the **Price Projection** is supposed to be some (linear) function of the *past* returns data, which is supposed to be a prediction of the *future* returns data (and hence prices). Any function of the past price data which is correlated with the future returns can constitute a valid **Price Projection**, so long as it makes use only of *past* data – no *future* data allowed! Then, for an *N*-day **time horizon**, we wish to compute the correlation between the **Price Projection** and the ***N*-day future returns**, summed over days –1 through –*N*. In order to do this, we need to use the formula for the *sample covariance* given in the Appendix, and divide the *covariance* by the *variance* to get the **correlation**. (The *correlation* always ranges between –1 and +1.) So we need to go back each day, so that each day in the past is "day 0", the *present* day "relative" to this *past* day, and compute a separate **Price Projection** using only data to the *past* of (and including) this previous "day 0". If (n) denotes the n'th day in the *past*, (with n a *positive* number or zero) then "day 0" relative to this n'th day in the past will be denoted by $0(n)$. Similarly, the day with index k relative to this n'th day in the past will be denoted by $k(n)$. So the indexes in the above table should be denoted by $k(0)$. We may similarly extend the table going back, say, 1024 days in the past, as follows:

…	**+4(0)**	**+3(0)**	**+2(0)**	**+1(0)**	**0(0)**	**–1(0)**	**–2(0)**	**–3(0)**	**…**
…	**+4(1)**	**+3(1)**	**+2(1)**	**+1(1)**	**0(1)**	**–1(1)**	**–2(1)**	**–3(1)**	**…**
…	**+4(2)**	**+3(2)**	**+2(2)**	**+1(2)**	**0(2)**	**–1(2)**	**–2(2)**	**–3(2)**	**…**
…	**+4(3)**	**+3(3)**	**+2(3)**	**+1(3)**	**0(3)**	**–1(3)**	**–2(3)**	**–3(3)**	**…**
…	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**	**…**

So, since ***QuanTek*** typically uses 2048 days of data (8 years), and the Price Projection uses 1024 days of data, we can go back 1024 days and compute a Price Projection for each of the 1024 days in the past, each using the previous 1024 days of data. So the table above will have 1024 rows.

It will have 1024 columns of *past* data (black), and since the Price Projection is computed 100 days into the future, it will have 100 columns of *future* Price Projection (blue). As a reminder, in the above table, day 0(0) is actually *day 0*, day 0(1) is actually *day +1*, day 0(2) is actually *day +2*, day 0(3) is actually *day +3*, and so forth, going back 1024 days in the past. Each Price Projection relative to *n* days in the past is computed using only *past* data relative to *n* days in the past. In the **Correlation Test**, when you compute the **Price Projection** 1024 times, what you are doing is filling in the blue areas of this table using data relative to *n* days in the past, for *n* ranging from 0 to 1023.

Now the sample **correlation** can be computed between the **Price Projection** and the **future returns**. To start, suppose we want to compute the **covariance** between the 1-day Price Projection (of the returns) and the actual 1-day future returns. This sample covariance will be the sum over the past 1024 days, of the product of the projected 1-day return with the actual 1-day return. We don't know the future 1-day return relative to day 0(0), because it hasn't happened yet. But relative to 1 day in the past, the 1-day future return relative to day 0(1) is the return 0(0). Similarly, the 1-day future return relative to day 0(2) is the return 0(1), and that relative to day 0(3) is the return 0(2). The *projected* return relative to day 0(1) is the projection –1(1), that relative to day 0(2) is the projection –1(2), and that relative to day 0(3) is the projection –1(3). So the sample covariance is the sum of the products of these 1-day projected returns with the corresponding actual returns, going back 1024 days in the past. Similarly, we can also compute the covariance with an *N*-day **time horizon**, by taking the sum of the first *N* days of the Price Projection, and computing the covariance with the corresponding *N*-day average of the future returns. Finally, the **covariance** is divided by the standard deviations to get the **correlation**. (This is the ordinary, *Pearson's R* definition of correlation. The two *robust* methods of calculating correlation can be calculated using the same data arrangement.)

However, this method is even more general than described so far. We may not be sure whether the *phase* of the **Price Projection** is correct. Also, we may want to compute the correlation between *past* returns and *future* returns. So instead of just computing the correlation between the 1-day Price Projection and 1-day future returns, we can compute the correlation between *any* day's past data or future projection, and the 1-day future returns. In other words, instead of just taking a sum of products of the projection with index –1(*n*) with the returns with index 0(*n* –1), we take a sum of products of the data for *any* *k*(*n*) with the returns with index 0(*n*

–1), summed over all *n* from 1 to 1023. The difference between the index *k* and the value –1 we denote as the **lead time**. A separate correlation is computed for a range of lead times, and this correlation as a function of lead time is displayed as a graph. The correlation corresponding to *zero* lead time appears under the vertical line marked ZERO on the graph. The graph can then be shifted back and forth, so that different lead times appear under the vertical line, using the **Lead Time** control. In this way, the entire 100 days' worth of filter output of the Price Projection can be tested for correlation with the 1-day future return, as can all the past returns (back to +1024 days or so). By taking *N*-day sums as described previously, the correlation can also be measured for an *N*-day **time horizon**, for any desired **lead time**. Each choice of **lead time** really amounts to a separate **technical indicator**, so in this way a tremendous number of distinct indicators can be tested all at once. In fact, some surprising correlations between the *past N*-day returns, for various time lags, and the *future N*-day returns, can be uncovered in this way using the **Correlation Test – LP Filters & Indicators** test. In this way, a large number of possible **technical indicators** can be tested, one for each possible setting of the **Lead Time** control, not only the one indicator (the **Price Projection** itself) corresponding to a **Lead Time** of zero.

1.8 Statistical Tests – Wavelet Routines

Two of the statistical tests on the **Statistical Tests** dialog are designed to test the correct operation of the **Wavelet** routines. These are the **Wavelet Analysis** dialog and the **Wavelet Variance** dialog. The **Wavelet Analysis** dialog displays the properties of the wavelet coefficients and smoothing, while the related **Wavelet Variance** dialog displays properties of the wavelet variance. Both of these dialogs display the properties for both the **Discrete Wavelet Transform (DWT)** and the **Maximal-Overlap Discrete Wavelet Transform (MODWT)** types of wavelet transforms. The properties are displayed graphically using data from any security in the portfolio. Both of these tests are diagnostic tests for the **Wavelet** routines, and do not pertain directly to trading in the portfolio. For more information on the wavelet routines, see the book by Donald B. Percival & Andrew T. Walden, <u>Wavelet Methods for Time Series Analysis</u> [PW], from which the ***QuanTek*** wavelet routines were adapted.

Wavelet Analysis

The main purpose of the **Wavelet Analysis** dialog is to test the wavelet routines and view the properties of the **wavelet coefficients** and **wavelet variance**. After choosing a security, you can view either the **wavelet coefficients** or the **wavelet variances**, each using either the **DWT** or **MODWT** type of wavelet transform. You can view each wavelet level separately, or the sum of all levels. There is also a switch between a display of the **multi-resolution analysis (MRA)** and the **shifted wavelet coefficients**. The **MRA** is a decomposition of the data into spectral octaves, while the **shifted wavelet coefficients** are the wavelet transform of the data, shifted in time to correspond to the original data. When the display type is **coefficients**, and viewing the **sum of all levels**, the display should be identical to the **returns** themselves, using either the **DWT** or **MODWT** filters. This is an important test of the filters. You can toggle the display to compare the **coefficients display** with the raw **returns**, or the **variance display** with the **variance** of the raw **returns**. Also, you can toggle the display between the data from the chosen security, and a set of **random data** that is generated each time it is toggled. In this way you can study how the display changes with the changes in the random data set.

When viewing the **variance** displays on any of the wavelet levels, there should be a close correspondence between the displays for the **DWT** or **MODWT** type of wavelet transform. There should also be a close correspondence between the variance from the **MRA** and that from the **shifted wavelet coefficients**. But these displays will not be exactly the same. Also, on any setting of the **wavelet level**, you can view different settings of the **Wavelet Smoothing**. Once again, the wavelet smoothing should be similar, but not identical, between **DWT** and **MODWT**, and between the **MRA** and the **shifted wavelet coefficients**. If these conditions are fulfilled, it means that both types of wavelet transform are working correctly.

Finally, for any setting of the **Wavelet Level**, and the **Variance Smoothing** set on **Level Average**, you can view the level average variance on each level, or the sum of all levels. On the **DWT** setting, this level average variance should be the same between the **MRA** and the **shifted wavelet coefficients**, and the sum of all the level variances should be 100%. But on the **MODWT** setting, these will be different, and will not add up to 100%. (The level average wavelet variance is not conserved by the MODWT.) You can also compare these levels with the raw **returns** variance, which is always 100% on each level setting.

Wavelet Variance

The main purpose of the **Wavelet Variance** dialog is to investigate properties of the **covariance matrix** derived from the **wavelet variance**. The **Wavelet Variance** or **Covariance**

Matrix is displayed on a graph using either the **DWT** or the **MODWT** type of wavelet filter. You can view the **Wavelet Variance** with a variety of time scales of **Smoothing**, listed in days. Also there is a control to adjust the **Shrinkage**, which is a mixture of the **Wavelet Variance** and its level average, for the purpose of smoothing. These might be used in a type of LP filter which estimates the covariance directly from the wavelet variance (not used at present).

When viewing the **Covariance Matrix**, this is estimated by taking a "signal" which is unity at time zero and zero otherwise, and taking its **MRA**, using either the **DWT** or the **MODWT** type of wavelet filter. Then the **MRA** on each level is multiplied by the wavelet variance of the data from the chosen security, using the **Smoothing** and **Shrinkage** settings, at any given time index. You can choose any specific **Wavelet Level**, or the sum of all levels, so you can view each wavelet level of the **MRA** of the signal individually. Finally the inverse wavelet transform is taken and the result displayed on a graph. This inverse transform is interpreted as the covariance distribution due to the wavelet variance of the security. There is a control to adjust the vertical scale of the graph. There is another control to move the signal to different time indexes, to observe how the covariance changes with the time index. Finally, as a check, you can select a **Level Wavelet Variance**, meaning the case where the wavelet variance is constant and the same on all levels. This should lead to no covariance, so the signal appears as a single sharp line with a height of 100%. (This may be hard to see until you move it away from the ZERO line using the **Relative Time Lag** control.)

The theory behind this is that if the wavelet variance of the security is constant and the same on all levels, this is equivalent to white noise and there is no correlation. However, if the variance varies on different wavelet levels, then this indicates the presence of correlation and leads to a covariance between elements of the time series at different time indexes. If the wavelet variance on different levels changes with the time index, this leads to a time-dependent covariance matrix. The variance on each wavelet level contributes a term to the covariance matrix, which can be viewed on the graph. If the wavelet variance is equal on all levels, all these terms add up to a single sharp line, otherwise the distribution is spread out and there is non-zero covariance between different time indexes. So this dialog provides an interesting display of the interplay between the time-dependent level wavelet variance of the returns series and the non-zero covariance that results from it.

It might be added that the variance computed from the **Wavelet transform**, just like the **Fourier transform**, is mostly stochastic noise. To get an estimate of the "true" variance requires some kind of smoothing procedure. For this same reason, an estimate of the covariance matrix directly from the wavelet variance results in a covariance matrix that is mostly stochastic noise. The **Smoothing** and **Shrinkage** can help, but in general this method is unreliable due to the stochastic nature of the measured wavelet variance. So this method of computing the covariance matrix for use in an LP filter was not successful – a better procedure is the use of an **Adaptive filter**.

1.9 Statistical Tests – Spectrum of Returns

Two of the statistical tests in ***QuanTek*** are displays of the **spectrum** of the returns. These two tests are based on the **Fourier Transform** and the **Wavelet Transform** of the returns data. The graph of the power spectrum based on the Fourier Transform is a standard test in Time Series Analysis called the **Periodogram**. Due to the **Wiener-Khinchin Theorem**, the **Periodogram** is the **Fourier Transform** of the **autocovariance function**. This means that if the original time series is completely random "white noise", then the smoothed Periodogram should be constant. Similarly, the graph of the power spectrum based on the **Wavelet Transform** is the Wavelet Transform of the autocovariance function. Any deviation from a constant spectrum is an indicator of **correlation** in the time series. This is the main purpose of studying the **spectrum** of the time series – to detect possible **correlation**.

Spectrum Periodogram

The **Periodogram** is a method for measuring the **spectrum** of a time series, in this case returns data. A description of this test can be found in many standard textbooks, such as Brockwell & Davis [B&D]. The **Periodogram** is basically a **Fast Fourier transform (FFT)** of the stock returns data (squared), and displays the squared amplitude of each frequency component, in steps of the lowest frequency, up to the **Nyquist frequency** (with period 2 days). The amplitude of each sine wave of a given frequency is squared to give the *spectral power* at that frequency, and the result is displayed as a graph of power versus frequency, to make the **Periodogram**. It should be noted that only the bottom half of the spectrum is displayed, corresponding to 512 values of the power spectrum out of the 1024 total. This is because we feel that the upper half of the spectrum is not very significant, since it corresponds to cycles with periods between 2 and 4 days, and with daily data these are probably just stochastic noise. In fact, filtering out just this narrow range of frequencies eliminates *half* of the noise power spectrum, so we feel this is a worthwhile noise reduction strategy.

For comparison, a second method of spectrum estimation, called the **Maximum Entropy** method [NR], is also displayed. This method relies on the **Standard Linear Prediction** filter. The LP filter coefficients are computed from the returns data by means of the Burg Order-Recursive algorithm, then the **Maximum Entropy** method estimates the spectrum from these coefficients. It can be seen that the results are pretty similar in both cases.

According to the standard theory of the **Periodogram**, it must be **smoothed** on some time scale. If it is left unsmoothed, the *standard error* of each Fourier component is roughly 100% of the amplitude of the component. After smoothing on a time scale of N days, the *standard error* of the *smoothed* Fourier component is roughly $1/\sqrt{N}$. The default smoothing is set at 6 days, but you can change to a wide range of smoothing time intervals and view the resulting smoothed **Periodogram**. Please consult a standard text on **Time Series Analysis** for an explanation of the necessity for smoothing the **Periodogram**, such as Brockwell & Davis [B&D].

There are many peaks and valleys in the observed spectrum, but unfortunately it is not possible to show conclusively that these are any different from a random result. To demonstrate this, the **Periodogram** can be viewed using only **random Gaussian** data, generated by a random-number generator. To view the random data, click the **Random** button. Each time you toggle this button, a new set of random data is generated, and displayed in the two windows. So you can repeatedly compare the returns data **Periodogram** with that of the **random Gaussian** data, with a new set of random data each time. It can be seen that the Periodogram with **random Gaussian** data also displays the same type of peaks and valleys, so it can be concluded that, whatever correlations are present in the returns data, they are usually buried in stochastic noise.

Also included in this dialog box are two standard statistical tests. The **Kolmogorov-Smirnov** test [NR] compares the spectral distribution of the **Periodogram** to a constant distribution. It then computes the **confidence level** that the **spectrum** is *different* from a constant distribution. This can be interpreted as the probability that the spectrum was *not* obtained from a random Gaussian distribution by random chance alone. It will be observed that, using the random Gaussian data, this confidence level ranges from 0% to 100%, and is distributed roughly equally over this range. This is what you would expect from a purely random result. The **Fisher's** test [B&D] computes the **confidence level** for a **periodic component** in the spectrum. This is used to determine the probability that an observed *cycle* in the data is *not* obtained from a random Gaussian distribution by random chance alone. It will likewise be seen that, using the random Gaussian data, this confidence level also ranges from 0% to 100%, and is also distributed roughly equally over this range. It would be interesting to run these two tests over a collection of security data files, and observe whether or not the distribution of the confidence levels of the two tests is still constant from 0% to 100%.

Spectrum Wavelet

This dialog box uses an alternative method of measuring the spectrum based on the **Discrete Wavelet Transform (DWT)**. Unlike the Fourier basis, which are "infinite" sine waves, but have a single, precise frequency, the **wavelets** are waves that are localized in both time and frequency, but not infinitely sharp in either. When decomposed in the **wavelet** basis, the **wavelet coefficients** of the stock returns, which are the amplitudes of each wavelet basis component making up the waveform of the returns, are functions of both frequency and time. More precisely, the frequency spectrum is divided up into *octaves*, which are multiples of two in frequency. So the lower end of the spectrum has more divisions into octaves than the higher end. This is appropriate, since we are mainly interested in the low frequency end of the spectrum anyway, as this is what determines the long-term trend of the prices. To compute the **wavelet spectrum**, therefore, the returns time series is decomposed into its **wavelet coefficients**, and then these are squared to give the power spectrum. Finally, this power spectrum is *averaged over time* within each frequency octave. The result is displayed in a graph, which shows the power spectrum in each octave. There are only 9 octaves in all, as opposed to 1024 frequency components in the case of the **Periodogram**. This may be viewed as an alternative method of *smoothing* the **Periodogram**, and thereby (hopefully) eliminating some of the stochastic noise.

When you first open the **Spectrum Wavelet** dialog, the yellow bars you see are the one-*standard error* bars for each octave. Notice that the error bars are much larger for the low frequency octaves, because there are correspondingly fewer data points in these octaves. When you select a stock, the **wavelet spectrum** is displayed. The top graph shows this spectrum, averaged over all 2048 days of data, whereas the bottom graph shows a different type of wavelet spectrum called the **Maximal-Overlap Discrete Wavelet Transform (MODWT)**, which is time averaged over only the latest *N* days. You can set this number of days *N* to any value from 1 to 2048. This shows how the **MODWT** spectrum changes as the time period over which it is averaged changes.

The **Spectrum Wavelet** dialog contains two versions of the **Chi-Square** test. The first test measures the **confidence level** that the *time-averaged* **wavelet spectrum** is not random. This test usually gives a result that the spectrum is close to random, since there are only 9 "bins". On the other hand, the second **Chi-Square** test measures the **confidence level** that the *full* **wavelet spectrum** is not random. This test is also expressed in terms of *standard deviations* away from the center of the (Gaussian) probability distribution, which is another way of expressing the **confidence level**. This almost always gives an amazing number of standard deviations away from randomness for the *full* **wavelet spectrum**. So either there is some correlation that is being averaged over in the *time average* after all, or else this result may be explained by the fact that the *returns* themselves do not obey a Gaussian distribution, but instead have "fat tails". We are still investigating this question, but it is a very interesting result nonetheless.

1.10 Statistical Tests – Correlation of Returns

QuanTek includes two tests for correlation of returns, between the returns of the same security (autocorrelation) or between the returns of different securities. The **Correlation – Returns** dialog displays a **scatter graph** of the returns between any two securities in the portfolio. Often the returns of different securities are strongly correlated, and this information can be useful in choosing securities for a diversified portfolio. (Generally, to achieve maximum diversification, you want to find securities that are *anti-correlated* with each other, so that when one goes down, the other is likely to go up.) The **Correlation – Auto & Cross** dialog displays the correlation between the two securities as a function of time lag. If the two securities are the same, then it displays the autocorrelation sequence.

Correlation – Returns

An interesting statistical display in ***QuanTek*** is the **Correlation – Returns** dialog, which is a **scatter graph** of the returns of two different securities. This is intended to display, graphically, the **correlation** or **anti-correlation** of the returns of the two securities. If the returns are correlated, then the dots of the scatter graph will tend to line up along the diagonal line of the graph, while if they are anti-correlated, they will tend to line up along the opposite diagonal. If the returns are uncorrelated, the display is designed so that the dots should be evenly distributed over the whole square area of the display (assuming a double exponential distribution of the returns). Connected with this display is a measure of the correlation, using three different methods. First there is the ordinary **Pearson's R** method of measuring correlation, which measures **linear correlation** [NR]. But there are also two different **nonparametric** or **rank-order correlation** methods, which are also called **robust** methods of measuring the correlation. These are the **Spearman Rank-Order Correlation** and **Kendall's Tau** [NR]. These robust methods do not depend on the random variables belonging to a Gaussian distribution (which they really do not – the distribution has "fat tails"), and hence they are less likely to indicate spurious correlation where no correlation really exists.

This **Correlation – Returns** dialog shows some rather strong correlations for certain pairs of stocks, especially those in the same sector of the economy. This correlation is useful to know from the standpoint of reducing **risk** in an **optimal portfolio**. If some of the securities in the portfolio are strongly correlated, then this increases the **market risk** because if one security loses value, all the securities that are correlated with it also tend to lose value. To reduce risk to a minimum, it is desirable to choose securities that are uncorrelated, or even anti-correlated, so that fluctuations in the value of one security will be hedged or compensated by fluctuations in the other securities in the portfolio.

Correlation – Auto & Cross

If you click the **Correlation** button in the **Correlation – Returns** dialog, the **Correlation – Auto & Cross** dialog box is displayed, which contains a set of bar graphs of the **correlation** between the two securities, as a function of **time lag**. The **time lag** is just the time difference, in days, between the returns that are compared in the correlation test. One graph for positive time lags and one graph for negative time lags are displayed. If you choose the same security for both, then you can view the **autocorrelation** of the returns of that security. In that case, the two bar graphs for positive and negative lags will be the same. If you change the **time horizon** for the correlation to *N* days, then the correlation between *N*-day returns is computed. In this case, the bars of the bar graphs become *N* pixels wide rather than just one pixel wide.

This **Correlation – Auto & Cross** dialog is useful for comparing the degree of correlation between different securities for the purpose of portfolio selection. It is also useful for general studies of the correlation structure of the stock returns data. For example, when studying the autocorrelation of daily returns, if you look closely you can see that the autocorrelation for the first three days of lag is almost always *negative*. Often you can also spot what look like cycles in the correlation structure, with periods in the intermediate-term range of, say, one to several months.

1.11 Statistical Tests in the Graph Window

In addition to the statistical tests in the **Statistical Tests** dialog, there are also several tests available directly in the **Graph Window** for a security. These make use of the historical data and **Adaptive filter** calculations for that security. The most important test is the **Price Projection** itself, as calculated by the **Adaptive filter** and displayed on the graph. Another test is the actual **Historical Error Bars**, which measure the actual difference between the historical **Price Projection** and the actual "future" prices. This is a measure of the actual accuracy of the **Price Projection**, which in turn is a measure of the **correlation** between the projected future returns and the actual future returns.

Historical Price Projection

When new data are downloaded, and the **Graph Window** is opened, you are asked if you want to update the **Adaptive filter** calculation. If so, then a new calculation is begun starting 2048 days in the past, in which time the **Adaptive filter** will "adapt" to the data and then display the result as a straight-line **Price Projection**. If not, then a quick calculation using the **Standard LP filter** is displayed instead. (This display is not in the form of a straight line.) If the **Adaptive filter** is up-to-date, you can then toggle the **Toggle Adaptive Filters** toolbar button to switch back and forth between the two filters, for comparison. You can also select an alternative LP filter using the **Select LP Filter** button, and calculate this alternative filter. Then the **Toggle Adaptive Filters** toolbar button toggles between the default **Adaptive filter** and the alternate **Adaptive filter**.

If you click the **Historical Price Projection** button, then you can select a date in the past using the Calendar Control, and the Graph will be displayed as it was on that day, including the **Price Projection** for that day. You can then compare the **Price Projection** for that day with the actual "future" prices. You can also use the **Toggle Adaptive Filters** toolbar button to toggle back and forth between the default **Adaptive filter** and either the **Standard LP filter** or the alternate **Adaptive filter**, for that day, as previously described. So by using the **Historical Price Projection**, you can see directly how the **Price Projection** due to the default **Adaptive filter** compares with the other filters and with the actual "future" prices. (To see the actual "future" prices, you need to select another date in the Date Calendar.)

Historical Error Bars

The error bars on the future **Price Projection** that are displayed when the **Graph Window** first opens are only an estimate. They are computed by measuring the average volatility, or high (log) price minus the low (log) price, and then extrapolating this into the future *n* days by multiplying it by $\sqrt{n+1}$, where *n* is the index of the future day. So, for example, for the first future day with $n=1$, the average volatility is multiplied by $\sqrt{2}$ and displayed as an error bar. This is just an estimate based on the properties of the **Random Walk** model. The error bars are centered on the future **Price Projection**.

You can compute the actual historical error bars, using the **Projected Error Bars** toolbar button. This goes back 1024 days (for a data set of 2048 days) and measures the difference between the actual historical price and the **Price Projection**, for each past day, going forward 128 days in the future. The default **Adaptive filter** is used for the **Price Projection**. The errors above the actual price and below the actual price are counted separately, for each of the 128 future days. Then the average is taken over the past data set and displayed as a set of actual error bars, one for each of the 128 future days. This can then be compared with the estimated error by toggling the **Projected Error Bars** toolbar button. This is another direct test of how well the **Adaptive Filter** is working. The narrower the actual error bars, the better the **Adaptive filter** is tracking the actual future price action. Also if there is any systematic error in the **Adaptive filter**, the upper and lower error bars will be different lengths, which will be immediately visible in the display. So this is another good test of the effectiveness of the **Price Projection** from the **Adaptive filter**.

Simulated Trading

This test is planned for a future (hopefully near future) version of ***QuanTek***. An older version incorporated a simulated trading test for a single security, using **Trading Rules** for that security alone. However, a much more pertinent test would be simulated trading for the entire portfolio. Starting with an updated **Adaptive filter** calculation for each security, which saves the expected return for the entire history of the security (up to 8 years), what would be required would be to step forward one day at a time and perform the **Optimal Portfolio** calculation each day, compute the **Buy/Sell** signals for that day, then record the trades for the entire historical time period. This would then provide an explicit example of **maximizing returns** while **minimizing risk** for the portfolio as a whole. (The main question is how much computer power and memory it would require.)

1.12 Appendix: Definition of Correlation

The standard definition of linear correlation of two random variables, called **Pearson's R**, is given by [NR]:

$$r \equiv \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$$

Here, $\bar{x}$ and $\bar{y}$ are the mean values of the two random variables. There are two other types of correlation, which are called **robust** correlations, which are the **Spearman Rank-Order** and **Kendall's Tau** [NR]. These are called nonparametric methods of computing correlation because, unlike the linear or Pearson's R correlation, they do not presuppose a *Gaussian* distribution of the random variables. The Spearman Rank-Order correlation is the linear correlation of the *ranks*, as opposed to the linear correlation of the values of the variables as in linear correlation. To compute the ranks, the values are arranged in increasing order, and the order of each value is its *rank*. *Kendall's Tau* uses the correlation of the numerical *order* of the ranks (greater than, less than, or the same), as opposed to the difference in *value* of the ranks as in *Spearman Rank-Order*. These two robust methods are more reliable when the distribution of the random variables is non-Gaussian, and in particular when the distribution has "fat tails" as is the case with most financial data.

However, for our purposes a modified definition of correlation is more suitable. The problem with the above definition is that it breaks down when the buy-and-hold strategy is considered. To be specific, one of the above random variables will represent the future returns, and the other will represent the *trading rules*, or amount to be invested in a short-term trading strategy. If *s* is the number of shares, and δp is the actual (not logarithmic) returns (change in price per share), then the expected (simple) gain *g*, in dollars, is given by (summed over the trading days in a given time interval):

$$\begin{aligned} s &\equiv \text{number of shares} \\ \delta p &\equiv \text{actual returns} \\ g &\equiv \sum_i s_i \delta p_i = \text{gain} \end{aligned}$$

For *y* in the correlation formula we may use the logarithmic returns as a conservative estimate for the actual returns:

$$y_i \equiv \ln\left(p_i + \delta p_i\right) - \ln p_i = \ln\left(\frac{p_i + \delta p_i}{p_i}\right) = \ln\left(1 + \frac{\delta p_i}{p_i}\right) \le \frac{\delta p_i}{p_i}$$

The amount invested, in dollars, at time *i* is given by:

$$d \equiv s \times p = \left(\#\text{shares}\right) \times \left(\text{price per share}\right) = \left(\text{dollar amount invested}\right)$$

Thus we have:

$$\begin{aligned} g \equiv \text{gain} &= \sum_i \left(s_i p_i\right) \frac{\delta p_i}{p_i} = \sum_i d_i \frac{\delta p_i}{p_i} \\ &= \left(\text{dollar amount invested}\right) \times \\ &\quad \times \left(\text{fractional change in price}\right) \end{aligned}$$

For the *annualized simple gain* we sum over the number of trading days in a year, assuming we are dealing with daily returns, which may be taken to be 256 days.

The *trading rules* variable *x* is defined as the dollar amount invested at any given time, relative to the average amount of equity invested over the time period. This average equity can be either long or short, so the average equity invested is given by the *average absolute value* of the dollar amount invested over the time interval:

$$x_i \equiv \frac{d_i}{\left\langle \left| d_i \right| \right\rangle} \equiv \text{"trading rules" variable}$$

Here we define the *average absolute value* of the dollar amount invested over the time interval by:

$$\langle |d_i| \rangle \equiv \frac{1}{N}\sum_i |d_i| \equiv \text{average absolute value of equity invested}$$

The average absolute value of the equity invested, as a percentage of the total equity available to invest, is called the *average margin leverage*. To compare measured correlations to measured returns from trading rules, we normalize the average margin leverage to 100%. In other words, the normalized gain, denoted $\tilde{g}$, will be given by the *annualized simple gain* divided by the *average absolute value* of the dollar amount invested.

However, the correlation is expressed in terms of the root mean square of the trading rules, not the average absolute value of the trading rules (which is defined to be unity). We need to convert between one and the other. This is straightforward if we assume the random variables are distributed according to a Gaussian distribution. Denoting a Gaussian random variable by z, with standard deviation σ, it is well known that the Gaussian distribution (assuming $N \to \infty$) is normalized as follows:

$$\int_{-\infty}^{+\infty} \exp\left[{-z^2}\big/{2\sigma^2}\right] dz = \sqrt{2\pi}\sigma$$

The r.m.s. value of z is then given as the square root of the mean value of z^2, which (the latter) is defined to be the variance:

$$\langle z^2 \rangle \equiv \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} z^2 \exp\left[{-z^2}\big/{2\sigma^2}\right] dz = \sigma^2 \equiv \text{variance}$$

The average absolute value of z, on the other hand, is given as follows:

$$\begin{aligned}\langle |z| \rangle &\equiv \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} |z| \exp\left[{-z^2}\big/{2\sigma^2}\right] dz = \frac{2}{\sqrt{2\pi}\sigma}\int_{0}^{\infty} z \exp\left[{-z^2}\big/{2\sigma^2}\right] dz \\ &= \frac{1}{\sqrt{2\pi}\sigma}\int_{0}^{\infty} \exp\left[{-z^2}\big/{2\sigma^2}\right] dz^2 = \frac{2\sigma^2}{\sqrt{2\pi}\sigma} = \sqrt{\frac{2}{\pi}}\sigma\end{aligned}$$

Thus we have the following general relationship between the average absolute value of a Gaussian variable and its standard deviation (root mean square value):

$$\sqrt{\langle z^2 \rangle} \equiv \sigma = \sqrt{\frac{\pi}{2}}\langle |z| \rangle \approx 1.2533\langle |z| \rangle$$

Thus when any quantity is normalized to unit average absolute deviation (dividing by the average absolute deviation), it will be about 25% greater than when it is normalized to unit standard deviation (dividing by the standard deviation).

Thus the annualized gain, normalized to unit margin (average absolute amount of dollars invested) will be given by:

$$\tilde{g} \equiv \frac{g}{\langle |d_i| \rangle} \approx 1.2533 \frac{g}{\sqrt{\langle d_i^2 \rangle}}$$

This may be rewritten using the definition of the gain given above (renormalizing d_i in numerator and denominator by dividing by the average absolute deviation):

$$\tilde{g} \approx 1.2533 \sum_i \frac{d_i}{\sqrt{\langle d_i^2 \rangle}} \frac{\delta p_i}{p_i} = 1.2533 \sum_i \frac{x_i}{\sqrt{\langle x_i^2 \rangle}} \frac{\delta p_i}{p_i}$$

We may now use the inequality given above to rewrite this in terms of the logarithmic returns y_i:

$$\tilde{g} \geq 1.2533 \sum_i \frac{x_i}{\sqrt{\langle x_i^2 \rangle}} y_i = 1.2533 \sqrt{\langle y_i^2 \rangle} \sum_i \frac{x_i y_i}{\sqrt{\langle x_i^2 \rangle} \sqrt{\langle y_i^2 \rangle}}$$

Taking into account that there are 256 trading days in a year, we find:

$$\tilde{g} \geq 1.2533 \sqrt{\langle y_i^2 \rangle} \sum_i \frac{x_i y_i}{\sqrt{\frac{1}{256} \sum_i x_i^2} \sqrt{\frac{1}{256} \sum_i y_i^2}} = 1.2533 \cdot 256 \sqrt{\langle y_i^2 \rangle} \sum_i \frac{x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

Let us denote the average volatility, by which we mean the r.m.s. value of the logarithmic returns, by σ:

$$\sigma \equiv \sqrt{\langle y_i^2 \rangle} \equiv \text{root mean square logarithmic volatility}$$

We may then define our modified correlation, as follows:

$$\tilde{r} \equiv \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}} \equiv \text{"modified" correlation coefficient}$$

In other words, the modified correlation is the regular correlation with the mean values of the variables not subtracted off.

The annualized gain, normalized to unit margin, is the expected dollar gain divided by the average absolute amount of dollars invested. It is thus given in terms of the quantities defined above by:

$$\tilde{g} \geq 1.2533 \cdot 256 \cdot \sigma \cdot \tilde{r}$$
$$\tilde{g} \equiv \text{annualized (simple) gain, normalized to unit margin}$$
$$\sigma \equiv \text{root mean square logarithmic volatility}$$
$$\tilde{r} \equiv \text{"modified" correlation coefficient}$$

Thus the expected annualized simple gain, normalized to *unit margin* (unit average absolute amount of equity invested) is approximately given by the modified correlation multiplied by the average (r.m.s.) daily volatility of returns, times the number of trading days in a year and a numerical factor.

Thus we see that the meaningful quantity for the estimation of trading returns is this modified correlation, computed as if the mean values of the variables were zero, rather than the standard definition of correlation. In the ideal case of daily returns that are constant, the trading rules would be simply a constant amount invested, and then the modified correlation between the trading rules and the returns would be 100%. On the other hand, according to the usual definition of correlation, the correlation would be indeterminate because the variance of both the trading rules and returns would be zero; both of these would be equal to their mean values, so there would be zero in both the numerator and denominator. If, as often happens, the trading rules are nearly constant, then there would be very small quantities in both the numerator and denominator, and the computed correlation would be dependent on minute variations in the trading rules, which has very little to do with actual investment gains or losses. The modified correlation, on the other hand, would register the gain or loss to be incurred from the nearly constant investment, so it is the appropriate measure of correlation to be employed here.

The usual routines for measuring correlation [NR] use the data with the means subtracted off, so these routines must be modified to eliminate this subtraction of the means, resulting in the formula for the modified correlation given above. The *theoretical return* is then computed as above, multiplying this *modified correlation* by the *r.m.s. (logarithmic) volatility*, times the number of trading days in a year and a numerical factor, which results in a number which is approximately the actual gain, for small values of the daily returns, and is always less than or equal to the actual gain (so it is a conservative estimate).

1.13 References

[B&D] Peter J. Brockwell & Richard A. Davis, Time Series: Theory and Methods, 2nd ed. Springer-Verlag, New York (1991)

[CLM] John Y. Campbell, Andrew W. Lo, & A. Craig MacKinlay (CLM),
The Econometrics of Financial Markets,
Princeton University Press, Princeton, NJ (1997)

[E&M] Robert D. Edwards & John Magee, Technical Analysis of Stock Trends, 6th ed.
John Magee Inc., Boston, Mass. (1992)

[AH] Andrew C. Harvey, Forecasting, Structural Time Series Models and the Kalman Filter
Cambridge University Press, Cambridge, UK (1989)

[Hay] Simon Haykin, Adaptive Filter Theory, 4th ed.
Prentice Hall, Upper Saddle River, NJ (2002)

[SN] Sheldon Natenberg, Option Volatility & Pricing,
McGraw-Hill, Inc., New York, NY (1994)

[PW] Donald B. Percival & Andrew T. Walden,
Wavelet Methods for Time Series Analysis
Cambridge University Press, Cambridge, UK (2000)

[EP1] Edgar E. Peters, Chaos and Order in the Capital Markets,
John Wiley & Sons, Inc., New York, NY (1991)

[EP2] Edgar E. Peters, Fractal Market Analysis,
John Wiley & Sons, Inc., New York, NY (1994)

[NR] William H. Press, Saul A. Teukolsky, William T. Vetterling, & Brian P. Flannery (NR),
Numerical Recipes in C, The Art of Scientific Computing, 2nd ed.
Cambridge University Press, Cambridge, UK (1992)

[Pr] Martin J. Pring, Technical Analysis Explained, 3rd ed.,
McGraw-Hill, Inc., New York, NY (1991)

[Va] Tonis Vaga, Profiting From Chaos
McGraw-Hill, Inc., New York, NY (1994)