Study of color chaos model and Time frequency analysis of S&P 500 and NASDAQ indexes

by

Prabaharan Sivashanmugam

A Project submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Department of Mathematics and Statistics)
in The University of Michigan, Dearborn
2016

Advisor:

Professor Frank Massey, Chair

THE UNIVERSITY OF MICHIGAN

ARTES · SCIENTIA · VERITAS

1817

Dedicated to my wife Raji Praba, my sons, Deepak Praba and Darshan Praba

ACKNOWLEDGEMENTS

Immeasurable appreciation and deepest gratitude for the help and support are extended to the following persons who is one way or another have contributed in making this study possible.

Prof. Paul Watta, University of Michigan - Dearborn, for introducing Gabor transformation concept to me and provided base foundtional insights on Gabor applications in engineering field.

Prof. Frank Massey, University of Michigan - Dearborn, for his guidance and coaching to complete the project.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

ABSTRACT

Study of color chaos model and Time frequency analysis of S&P 500 and NASDAQ indexes

by

Prabaharan Sivashanmugam

Advisor: Professor Frank Massey

Change the abstract based on the finding of the report.. as of now, it is replication of the Chen's abstract Time frequency model and random walk model are two polar models in linear systems. Color chaos is a model that is in between these models which generates irregular oscillation with a narrow frequency band. The deterministic component from noisy data can be recovered by time variant filter in Gabor space. The characteristic frequency is calculated by Wigner decomposed distribution series. It is found that 7% of the detrended by HP filter can be explained by the deterministic color chaos. The existence of persistent chaotic cycle reveals a new perspective of market resilience and new sources of economic uncertainties. The nonlinear pattern in the stock market may not be wiped out by the market competition under non-equilibrium situations with trend evolution and frequency shifts.

CHAPTER I

Introduction

1.0.1 Objective of the project

The object of the project is to study the color chaos model that was paper published by Prof. Ping Chen in the paper [?]. Approach of the paper is to understand the mathematical concept of the color chaos, time frequency analysis - Wigner distribution, Gabor transformation, In the paper [?], the S&P 500 composite monthly index price was taken for the analysis, and in this project, the color chaos model is studied for both S&P 500 and NASDAQ index.

1.0.2 Organization of Project

This report has been organized into five chapters. Chapter 1 outlines the entire project giving an introduction to time series forecasting using stochastic models. Chapter 2 introduces color-chaos model and its applications. Chapter 3 provides an over view of time-frequency analysis. Chapter 4 Finish up after the chapters are finalized

1.0.3 Data Source:

Color chaos model is evaluated for two indices in the study and the source of the data for the indices are given below in table 1.2.

Symbol	Description	Source	Frequency	Duration
FSPCOM	S&P 500 Price Composite index	Citibase	Monthly	1942-1992
NASDAQ	NASDAQ Composite index	Yahoo Finance	Monthly	1970-2010

Table 1.1: Meta data on FSPCOM(S&P 500) & NASDAQ

1.0.4 Time series forecasting using stochastic models

In general, models for time series data can have many forms and represent different stochastic processes. The most widely used linear time series models are Auto Regressive (AR) and Moving Average (MA) models. Combining these two, the Autoregressive and Moving average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) are also used. Autoregressive Fractionally Integrated Moving Average (ARFIMA) model generalized ARMA and ARIMA models. For seasonal time series forecasting, a variation of ARIMA, the Seasonal Autoregressive Integrated Moving Average (SARIMA) model is used.

Linear models have drawn much attention due to their relative simplicity in understanding and implementation. Many practical time series show non-linear patterns. Non-linear models are appropriate for predicting the volatility changes in economic and financial time series. Considering these facts, various non linear models have been proposed over the years. A few widely used non-linear models are Autoregressive Conditional Heteroskedasticity (ARCH) model, its variations like Generalized ARCH (GARCH), Exponential GARCH (EGARCH), the Threshold Autoregressive (TAR), the non-linear Autoregressive (NAR), the non-linear Moving average (NMA) model and others.

The Autoregressive Moving Average (ARMA) models: An ARMA(p,q) model is a combination of AR(p) and MA(q) models and is suitable for univariate time series modeling. In an AR(p) model the future value of a variable is assumed to be a linear combination of p past observations and a random error together with a constant term.

Mathematically the AR(p) model can be expressed as:

$$y_i = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \epsilon_t \tag{1.1}$$

Here y_t and ϵ_t are actual value and random error respectively at time period t, $\varphi_i(i = 1, 2, 3, ..., p)$ are model parameters and c is a constant. The integer constant p is known as the order of the model. Sometimes the constant term is omitted for simplicity. AR(p) model regress against past values of the series, an MA(q) model uses past errors as the explanatory variables. The MA(q) model is given by

$$y_i = \mu + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t \tag{1.2}$$

Here μ is the mean of the series, $\theta_j(j = 1, 2, 3, ...q)$ are the model parameters and q is the order of the model. The random shocks are assumed to be a white noise process, i.e. a sequence of independent and identically distributed (i.i.d) random variables with zero mean and a constant variance σ^2. Generally, the random shocks are assumed to follow the typical normal distribution. Thus conceptually a moving average model is a linear regression of the current observation of the time series against the random shocks of one or more prior observations. Fitting an MA model to a time series is more complicated than fitting an AR model because in the former one the random error terms are not fore-seeable. Autoregressive (AR) and moving average (MA) models can be effectively combined together to form a general and useful class of time series models, known as ARMA models. Mathematically an ARMA(p, q) model is represented as:

$$y_i = c + \epsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{1.3}$$

Here the model orders p, q refer to p autoregressive and q moving average terms.

1.0.5 Difference Stationary

Loosely speaking a stationary process is one whose statistical properties do not change over time. More formally, a strictly stationary stochastic process is one where given $t_1, ...t_l$ the joint statistical distribution of $X_{t_1}, ...X_{t_l}$ is the same as the joint statistical distribution of $X_{t_1+\tau}$ for all l and τ. It means that all moments of all degrees (expectations, variances, third order and higher) of the process anywhere are the same. It also means that the joint distribution of (X_t, X_s) is the same as $(X_{t+\tau}, X_{s+\tau})$.

A stochastic process is said to be stationary if its mean and variance are constant over time. i.e. time invariant. A stationary process will not drift too far away from its mean value because of the finite variance.

If the trend in a time series is a deterministic function of time, such as t or t^2, we call it a deterministic (predictable) trend. If it is not predictable, we have a stochastic trend.

Consider the following model.

$$Y_t = \alpha + \beta_1 t + \beta_2 Y_{t-1} + u_t \tag{1.4}$$

where u_t is white noise.

Pure Random Walk: $\alpha = 0$, β_1=0, and β_2=1. This is non stationary as we get $Y_t = Y_{t-1} + u_t$. If we find the difference, we get $\Delta Y_t = u_t$. Note that differenced series is stationary (DS) because $E(\Delta Y_t) = E(u_t) = 0$ and $Var(\Delta Y_t) = Var(u_t) = \sigma^2$. Both are time invariant. Hence, a random walk without a drift is difference-stationary(DS).

Random Walk with a drift: $\alpha \neq 0, \beta_1 = 0$, and β_2=1. This is non stationary as we get $Y_t = Y_{t-1} + u_t + \alpha$. If we find the difference, we get $\Delta Y_t = \alpha + u_t$. Note that differenced series is stationary (DS) because $E(\Delta Y_t) = E(\alpha + u_t) = \alpha$ and $Var(\Delta Y_t) = Var(u_t) = \sigma^2$. Both are time invariant. Hence, a random walk with a drift is also difference-stationary(DS). Y_t is trending upward or downward depending on the sign of the drift (α) but this will be called a stochastic trend.

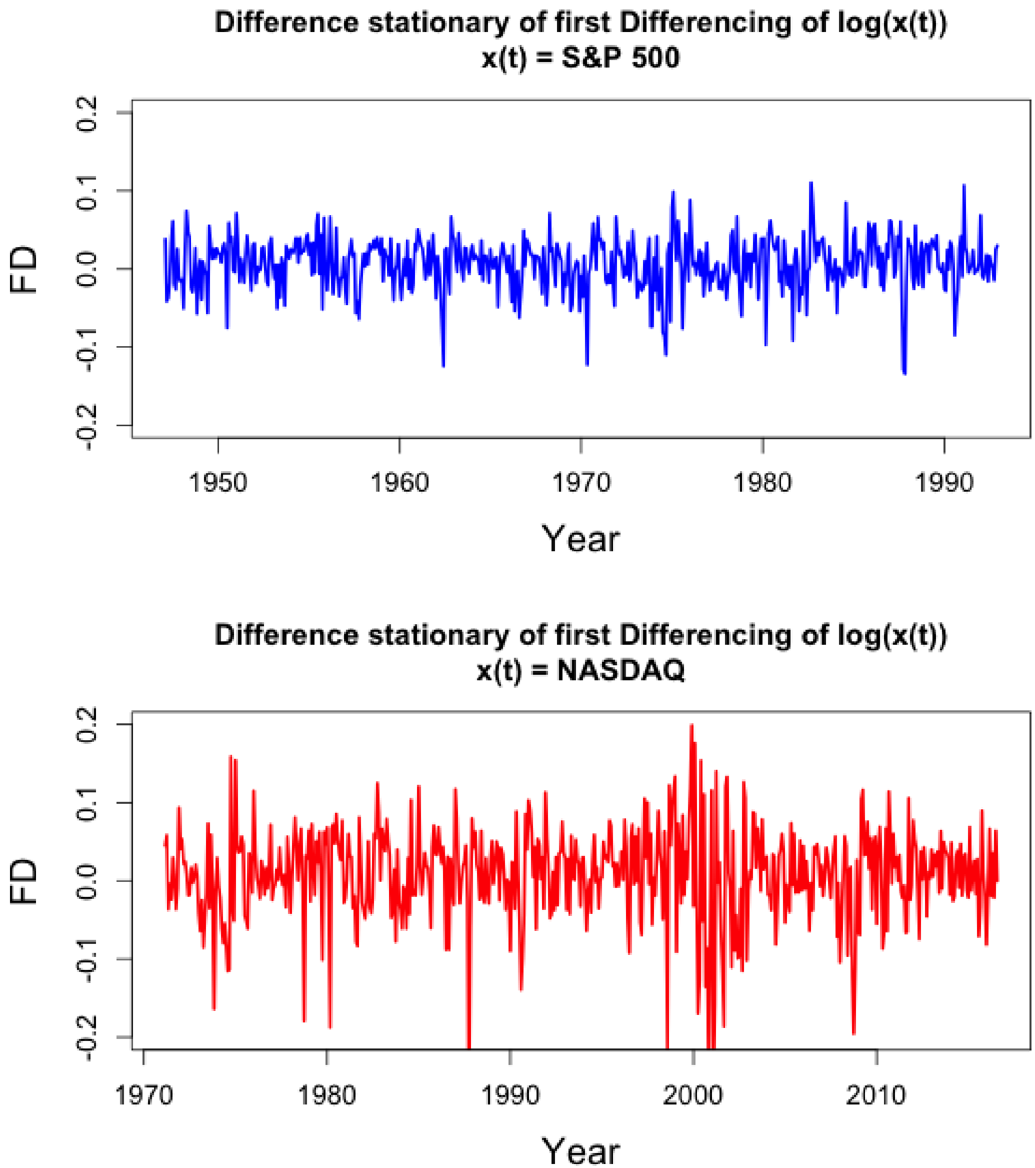


Figure 1.1: Difference stationary of natural log a) SP500 b) NASDAQ.The difference stationary [sp500(t2)-sp500(t1) and nasdaq(t2)-nasdaq(t1)] provides insights on the variation of signal. The R program fdplot.R used to generate the graph is given in Appendix A

Deterministic Trend: $\alpha \neq 0, \beta_1 \neq 0$, and $\beta_2 = 0$. Note that the mean of the series, $E(Y_t) = E(\alpha + \beta_1 t) = \alpha + \beta_1 t$, which is time-varying but its variance,

$Var(\Delta Y_t) = Var(\alpha + \beta_1 + u_t) = \sigma^2$ which is time-invariant. Still, the series with a deterministic trend is non-stationary. Once we know the values of α and β_1, we can subtract the mean from the series (detrending) and create a detrended series which is stationary.

Randow walk with drift and deterministic trend: $\alpha \neq 0, \beta_1 \neq 0$, and $\beta_2 = 1$. We get $Y_t = \alpha + \beta_1 t + Y_{t-1} + u_t$. Note that the difference series, $\Delta Y_t = \alpha + \beta_1 t + u_t$ is still time varying and hence, the mean of the differenced series is nonstationary. Detrending is still necessary on the differenced series to make it stationary.

1.0.6 Seasonal Trend Decomposition Procedure based on LOESS (STL)

STL is a filtering procedure for decomposing a time series into trend, seasonal, and remainder components. STL has a simple design that consists of a sequence of applications of the LOESS (LOcal regrESSion) smoother; the simplicity allows analysis of the properties of the procedure and allows fast computation, even for a long time series and large amount of trend and seasonal smoothing. Other features of STL are specification of amounts of seasonal and trend smoothing that range, in a nearly continuous way, from a very small amount of smoothing to a very large amount; robust estimates of the trend and seasonal components that are not distorted by divergent behavior in the data.

$$Y_t = f(S_t, T_t, E_t) \tag{1.5}$$

where Y_t is time series data at time t, S_t is seasonal component at time t, T_t is trend component at time t and E_t is remainder (or error or irregular) component of data at time t.

Additive decomposition is given by:

$$Y_t = S_t + T_t + E_t \tag{1.6}$$

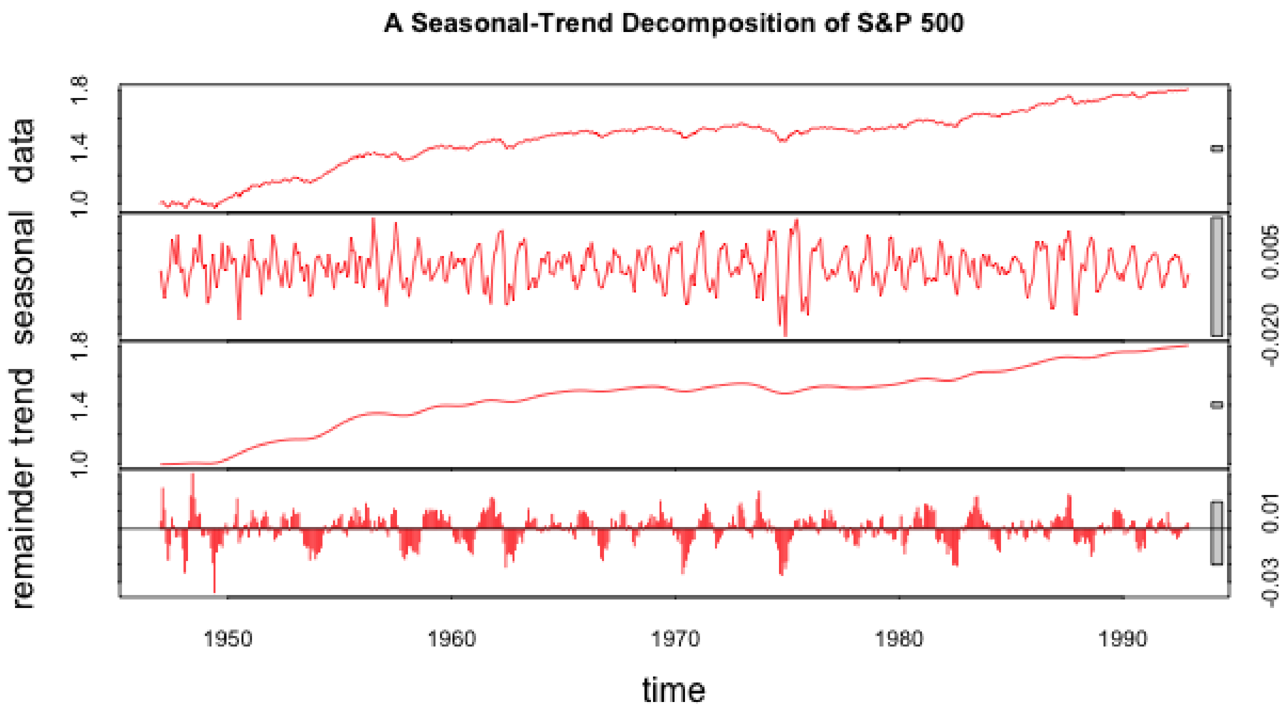


Figure 1.2: STL of Log SP500. a) The natural log of the SP500 b) The cyclical (or called seasonal) pattern c) The business trend of log SP500 d) Noise (remainder) data. The graph was created using the R program named as llt.R and it is attached in the Appendix A.

User controls the varitions on the trend and seasonal components.

In the 1.2, trend, seasonal and noise or error remainder data of SP500 are extracted using the seasonal trend decomposition procedure. The seasonal window ($value = 5$) used to control the variation on seasonal component.

In the figure 1.3, trend, seasonal and noise or error remainder data of NASDAQ are extracted using the seasonal trend decomposition procedure. The seasonal window ($value = 5$) used to control the variation on seasonal component.

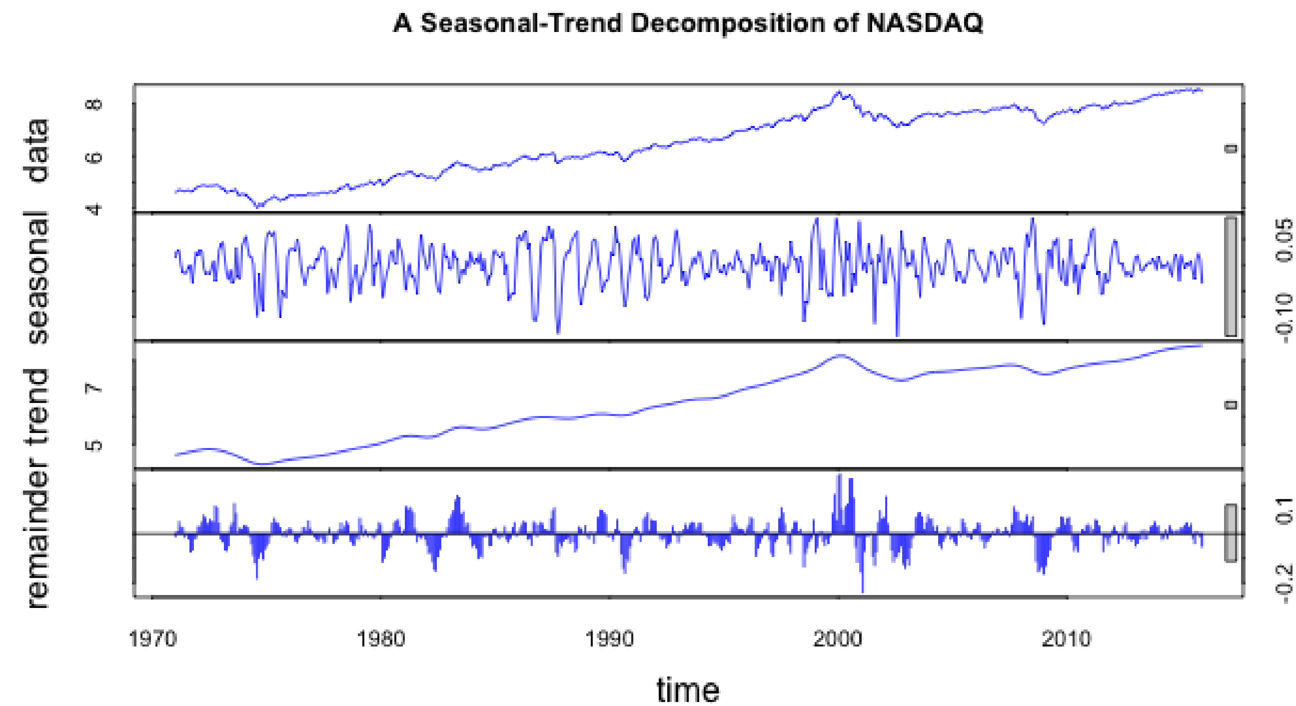


Figure 1.3: STL of NASDAQ. a) The natural log of the NASDAQ b) The cyclical (or called seasonal) pattern c) The business trend of log NASDAQ d) Noise (remainder) data.he graph was created using the R program named as llt.R and it is attached in the Appendix A

1.0.7 Log-Linear Method

When natural log values used for dependent variable and independent variable in its original scale, those models are called log linear models.

The following model of value in a savings fund that depends on initial investment, growth rate and time duration in which the funds are invested.

$$Y_t = Y_0(1 + r)^t \tag{1.7}$$

Where Y_t represents the value of the fund at the time t, Y_0 is the initial investment in the saving fund, and r is the growth rate.

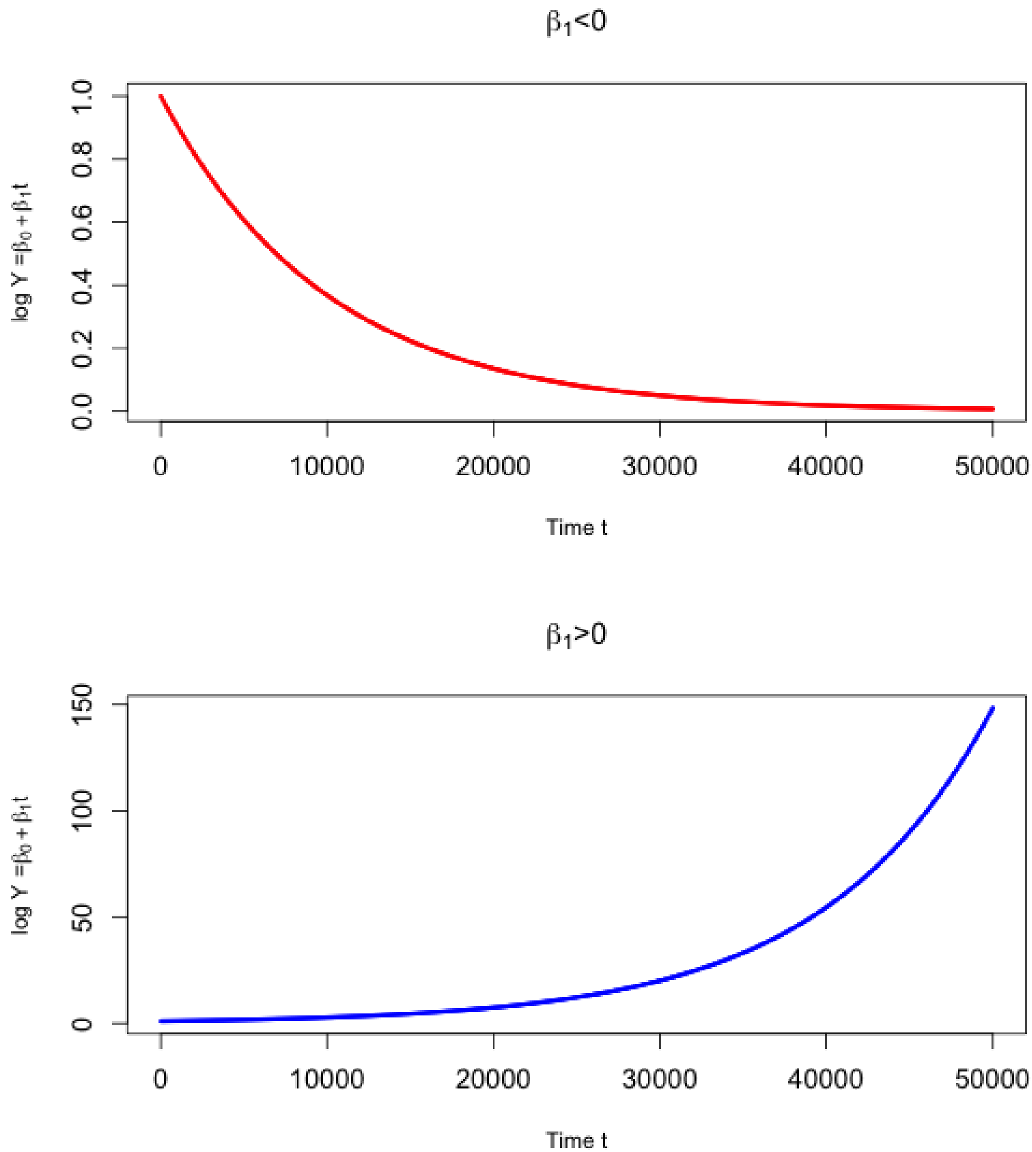


Figure 1.4: Log Linear Model when $\beta_1 < 0$ and when $\beta_1 > 0$. The graph was generated using loglinear.R, and it is attached in the Appendices A

$$logY = logY_0 + tlog(1 + r) \tag{1.8}$$

Where $logY_0$ is a constant and be β_0 and $log(1+r)$ be β_1 and $logY$ is given below:

$$logY = \beta_0 + \beta_1 t \tag{1.9}$$

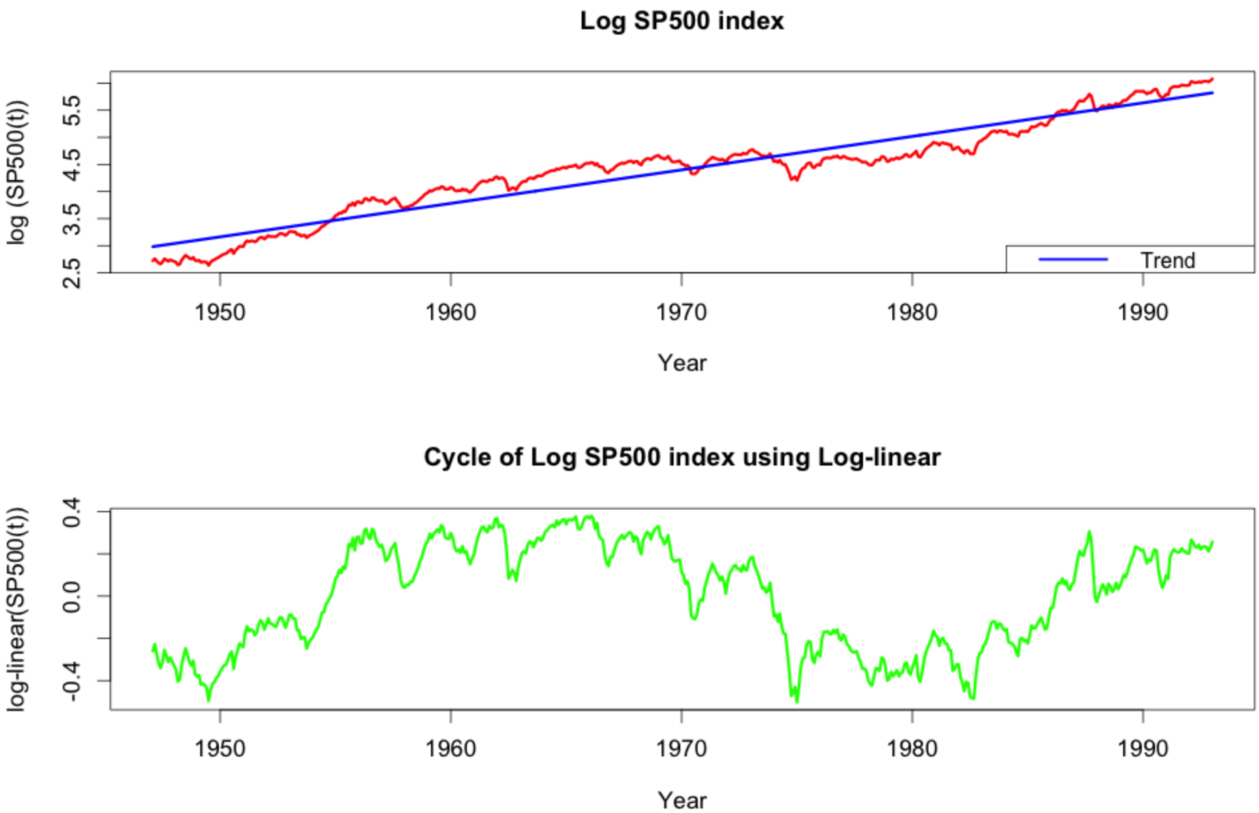


Figure 1.5: Log linear of trend and cycle for log SP500 a) Log SP500 and trend. The trend is calculated based on estimation of slope and y-intersect b)It is cycle of log-linear of SP500 and it is the variance of original log(sp500) and estimated trend. AutoCorrelation.R is the R program used to created the graph and it is attached in the Appendix A

After estimation of log-linear model, the coefficients can be used to determine the impact of the independent variables (t) on the dependent variable (Y). The coefficents in a log-linear model represent the estimated percent change in dependent percent change in dependent variable for a unit change in independent variable. The coefficient β_1 provides the instantaneous rate of growth. The regression coefficents in log-linear model does not represent the slope.

When $\beta_1 > 0$, the log-linear function illustrates a positive impact from the independent variable and when, $\beta_1 < 0$, the log-linear function depicts a negative impact

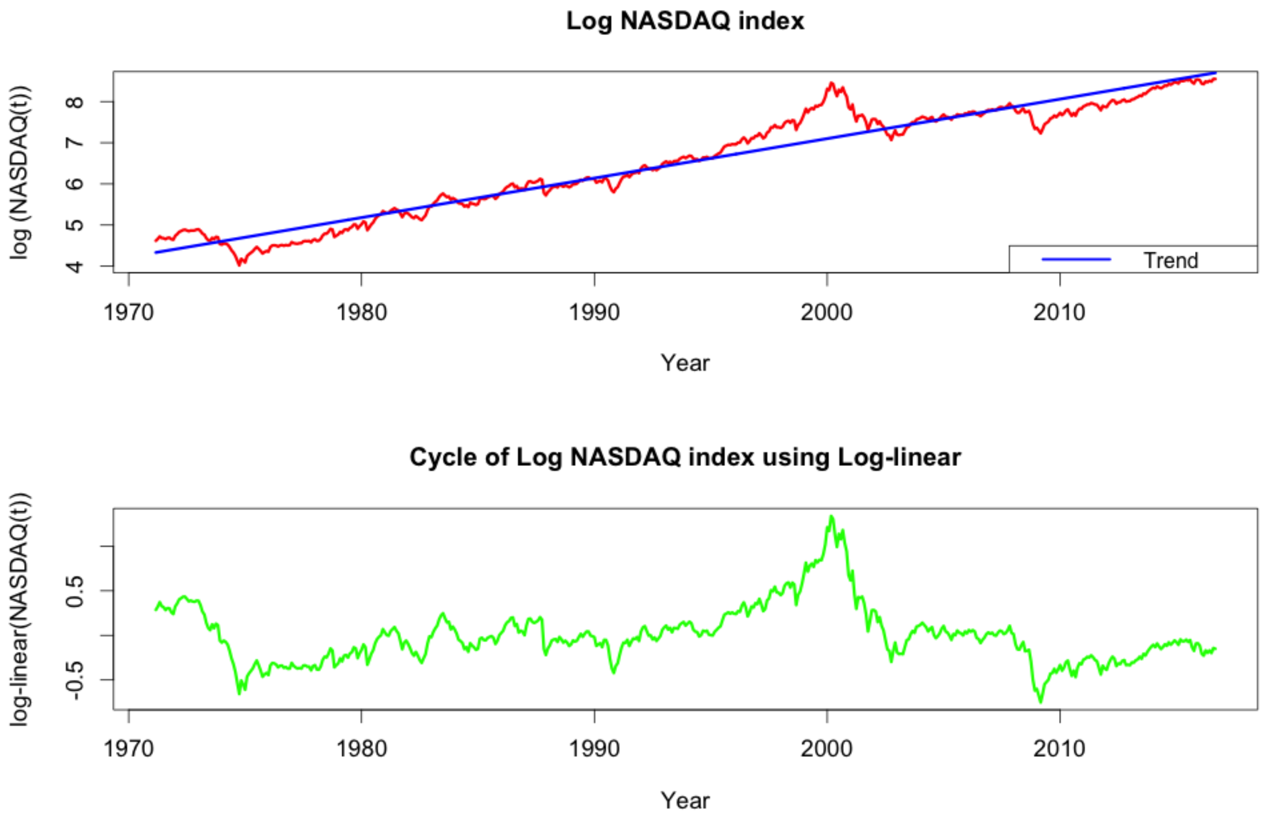


Figure 1.6: Log linear of trend and cycle for log NASDAQ a) Log NASDAQ and trend. The trend is calculated based on estimation of slope and y-intersect b)It is cycle of log-linear of SP500 and it is the variance of original log(NASDAQ) and estimated trend. AutoCorrelation.R is the R program used to created the graph and it is attached in the Appendix A

from the independent variable.

1.0.8 Auto correlation Functions

An autoregressive model is when a value from a time series is regressed on previous value from that same time series. For exampe, y_t on y_{t-1}:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t \tag{1.10}$$

In this regression model, the response variable in the previous time period has become the predictor and the errors have the same assumptions about errors in a simple linear regression model. The order of an autoregression is the number of immediate preceding values in the series that are used to predict the value at the present time. So, the preceding model is a first-order autoregression, written as $AR(1)$.

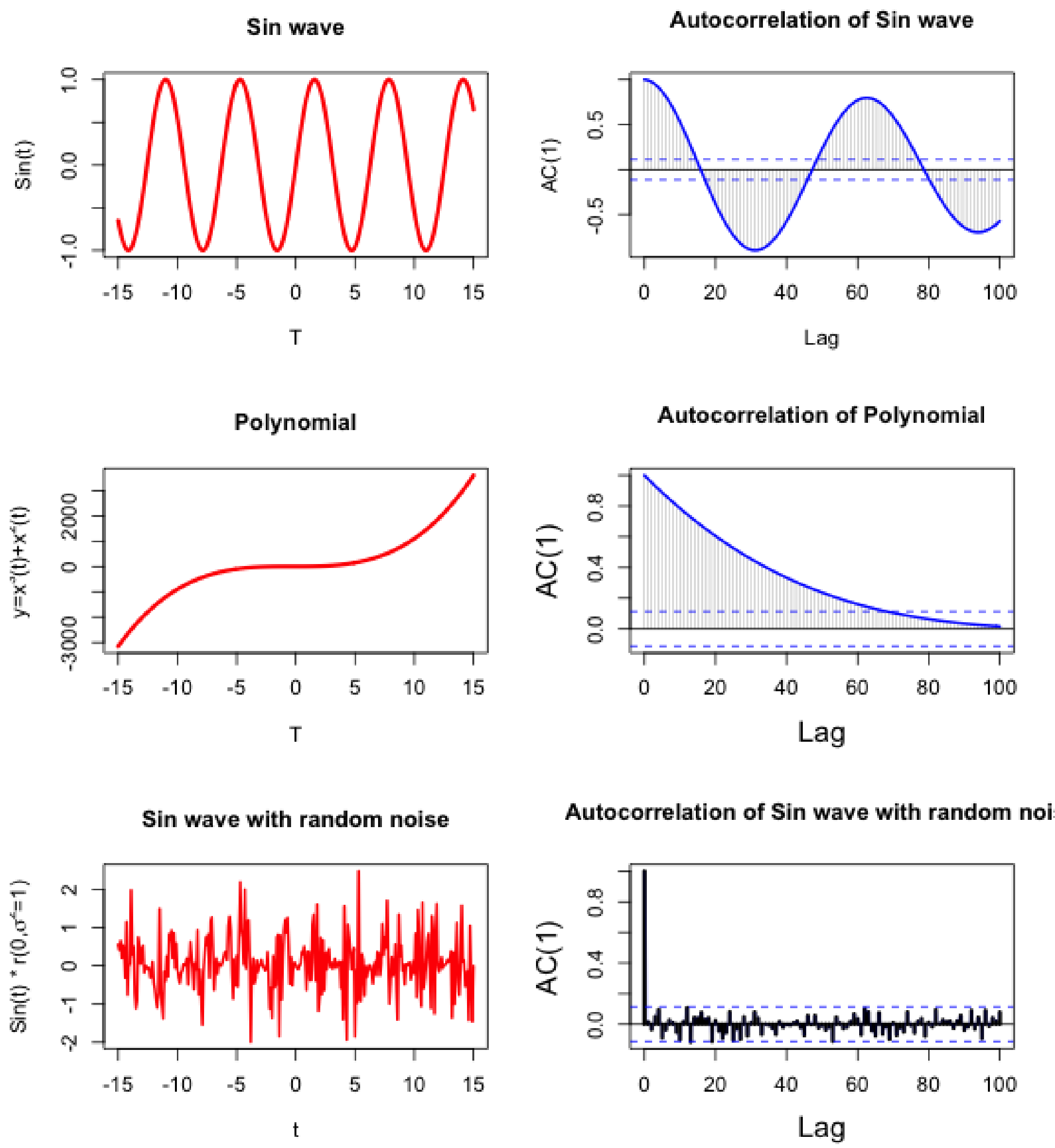


Figure 1.7: Auto correlation for a) Sin wave b) Polynomial equation c) Sin wave with random noise. AutoCorrelation.R is the R program used to created the graph and it is attached in the Appendix A

Let us say, if we want to predict y this year (y_t) using measurement of global temperature in the previous two years (y_{t-1}, y_{t-2}), then the autoregressive model for doing so would be:

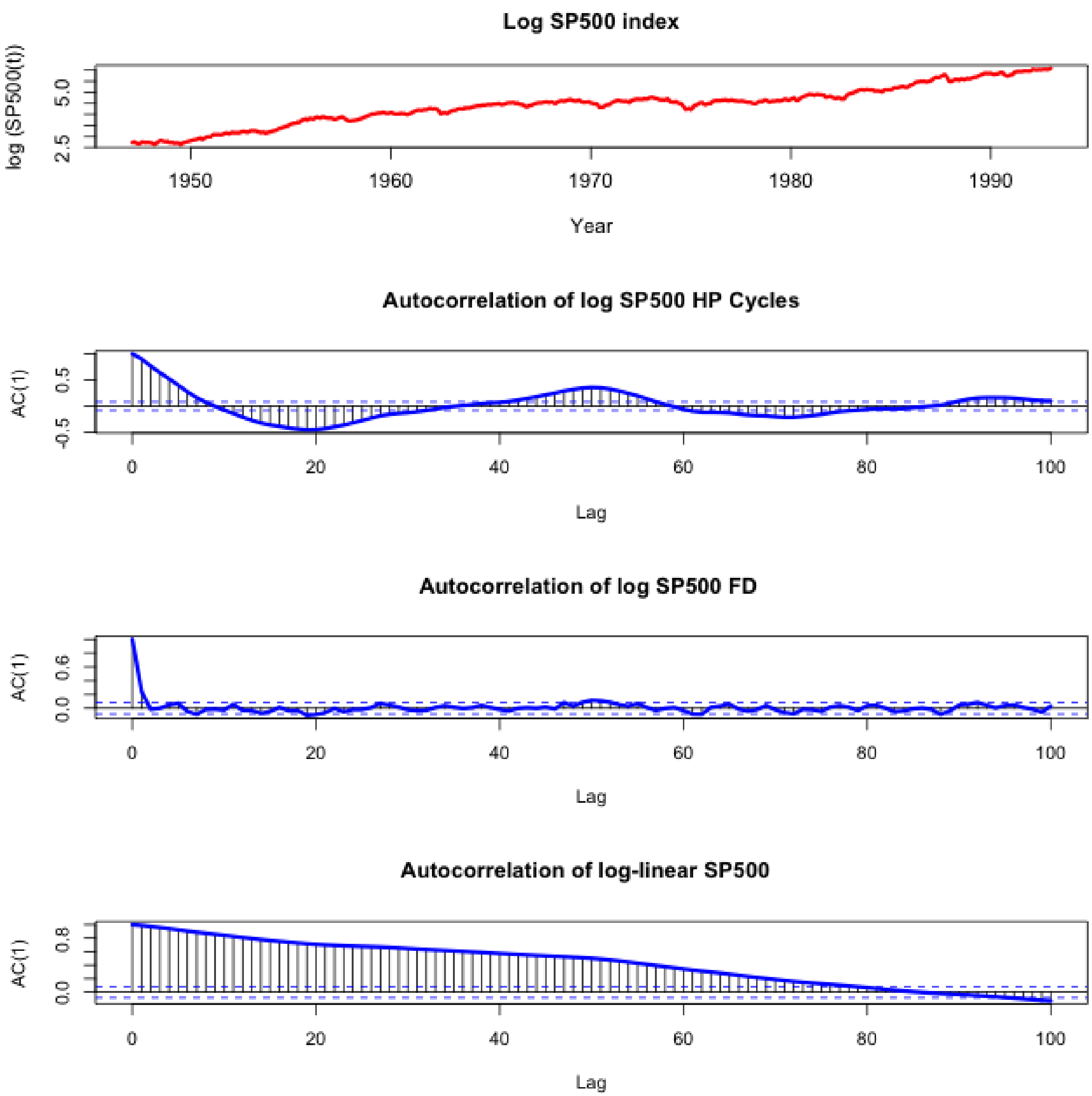


Figure 1.8: Auto correlation for log SP500 a) Log SP500 b)AutoCorrelation of HP cycle c) Autocorrelation of First Differencing d) Autocorrelation of log-linear SP500. AutoCorrelation.R is the R program used to created the graph and it is attached in the Appendix A

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \tag{1.11}$$

The above model is a second-order autoregression, written as $AR(2)$, since the value at time t is predicted from the values at times $t-1$ and $t-2$. More generally, k^{th} order autoregression, written as $AR(k)$, is a multiple linear regression in which the value of the series at any time t is a linear function of the values at times $t-1, t-2, ...t-k$.

The cofficient of correlation between two values in a time series is called the **autocorrection function** (ACF)

1.0.9 Hodrick and Prescott (HP) filter

Hodrick and Prescott proposed the HP filter to decompose a macroeconomic time series into a non-stationary trend component and a stationary cyclical residual component. The filter has become popular in applied macro economics in the last 15 years. Given an observed series y_i, let $y_i = x_i + c_i$, with $y^T = (y_1, y_2..., y_N), x^T = (x_1, x_2, ...x_N)$ and $c^T = (c_1, c_2...., c_N)$ where x_t denotes the unobserved trend component at time t and c_t the unobserved cyclical residual at time t. The HP trend $\hat{x}$ can be obtained as the solution to the following convex minimization problems:

$$\min_{[x_t]_{t=1}^N} \left[\sum_{t=1}^N (y_t - x_t)^2 + \lambda \sum_{t=2}^{N-1} ((x_{t+1} - x_t) - (x_t - x_{t-1}))^2 \right] \tag{1.12}$$

Here, λ is usually known as the smoothing parameter. As λ becomes larger, the HP estimated trend curve becomes smoother. The term being squared in the second sum of the equation, $(x_{t+1} - x_t) - (x_t - x_{t-1})$, or $\triangle^2 x_i$, is an approximation to the second derivate of x at time t. There are two opposing forces in the HP minimization problem. One force is attempting to minimize the sum of squared cyclical residuals and the other force is attempting to minimize the sum of squared

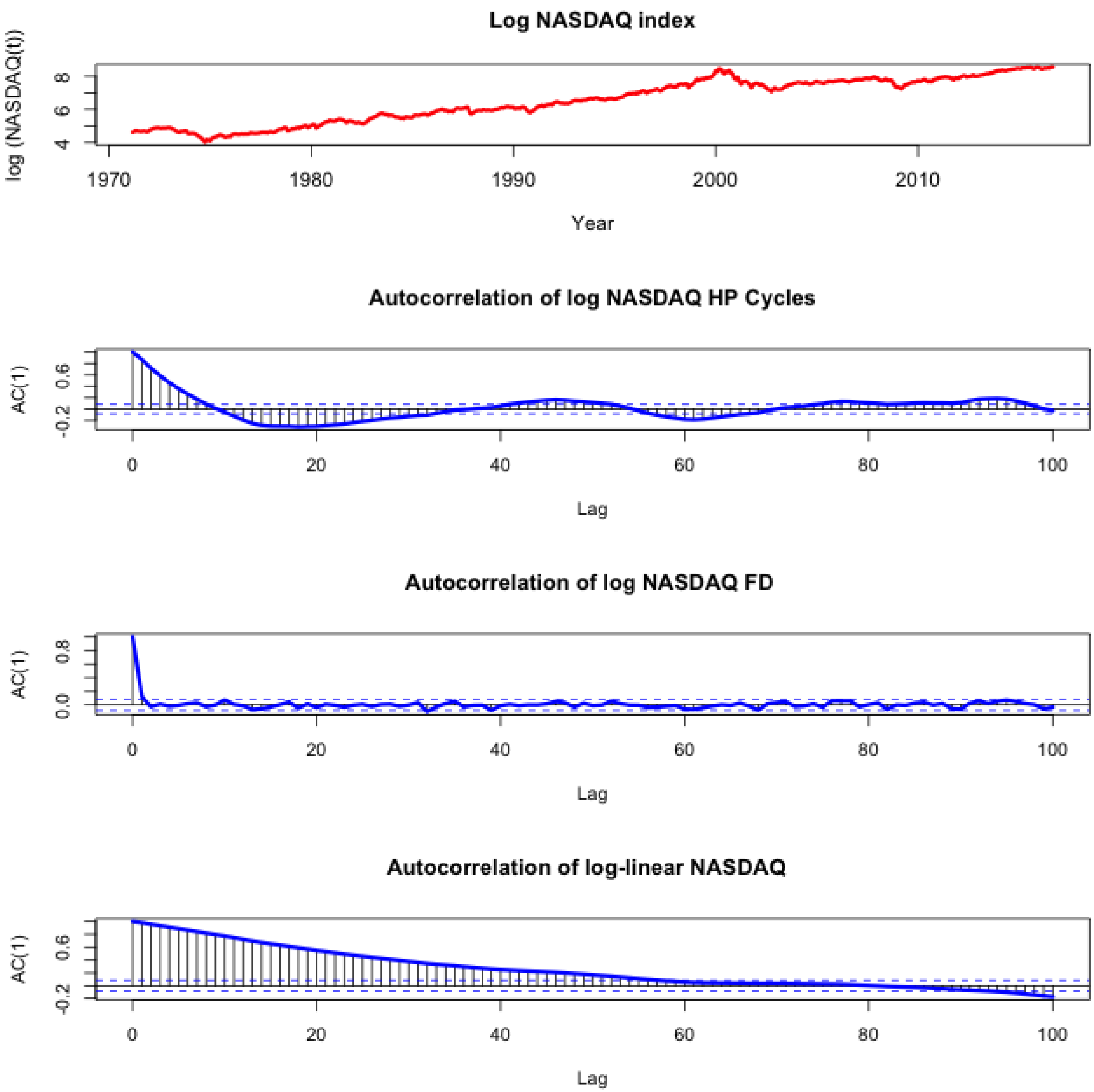


Figure 1.9: Auto correlation for log NASDAQ a) Log NASDAQ b)AutoCorrelation of HP cycle c) Autocorrelation of First Differencing d) Autocorrelation of log-linear NASDAQ. AutoCorrelation.R is the R program used to created the graph and it is attached in the Appendix A

$\triangle^2 x_i$. The smoothing parameter, λ, gives relative weight to these two opposing forces.

The λ parameter determines the smoothness of the trend component. The larger the value of λ, the higher the penalty in the second term. An empirical study indicates that a 5% cyclical component is moderately large, as is a 1/8th of 1% change in the

Symbol	Description	Source	Frequency	Duration
FSPCOM	S&P 500 Price Composite index	Citibase	Monthly	1942-1992
NASDAQ	NASDAQ Composite index	Yahoo Finance	Monthly	1970-2010

Table 1.2: Meta data on FSPCOM(S&P 500) & NASDAQ

Detrending	Mean	SD	Variance	T_0 (month)	P_{dc} (year)
FD	0.011	0.1123	0.0126	1.94	NA
HP	0.0083	0.2686	0.0721	8.93	NA
LLD	0.4265	0.3265	0.1065	86.6	NA

Table 1.3: Detrend Statistics on S&P 500

Detrending	Mean	SD	Variance	T_0 (month)	P_{dc}(year)
FD	0.0084	0.1070	0.0114	1.94	NA
HP	0.0094	0.1128	0.0127	8.93	NA
LLD	0.2583	0.3150	0.0992	86.6	NA

Table 1.4: Detrend Statistics on NASDAQ

growth rate in a quarter. In the paper [**?**], λ should be adjusted by multiplying it with the fourth power of the observation frequency ratios. This yields an HP parameter value of 6.25 for annual data given a value of 1600 for quarterly data.

1.0.9.1 First-order conditions of the HP minimization problem

The following HP first order conditions are derived by setting the gradient vector of the above minimization equation equal to zero. The first-order conditions are:

$$
\begin{aligned}
c_1 &= \lambda(x_1 - 2x_2 + x_3) \\
c_2 &= \lambda(-2x_1 + 5x_2 - 4x_3 + x_4) \\
c_t &= \lambda(x_{t-2} - 4x_{t-1} + 6x_t - 4x_{t+1} + x_{t+2}), t = 3, 4, 5, ..., N-2 \\
c_{N-1} &= \lambda(x_{N-3} - 4x_{N-2} + 5x_{N-1} - 2x_N) \\
c_N &= \lambda(x_{N-2} - 2x_{N-1} + x_N)
\end{aligned}
$$

or more compactly,

$$\boldsymbol{c} = \boldsymbol{\lambda F x}$$

where F =

$$\begin{bmatrix}
1 & -2 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & & 0 \\
-2 & 5 & -4 & 1 & 0 & \cdots & \cdots & & & 0 \\
1 & -4 & 6 & -4 & 1 & 0 & \cdots & & & 0 \\
0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & & 0 \\
0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 \\
\vdots & & & & & & & & & \vdots \\
\vdots & & & & & & & & & \vdots \\
\vdots & & & & & & & & & \vdots \\
0 & \cdots & \cdots & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\
0 & \cdots & & \cdots & 0 & 1 & -4 & 6 & -4 & 1 \\
0 & \cdots & \cdots & & & 0 & 1 & -4 & 5 & -2 \\
0 & \cdots & \cdots & \cdots & & & 0 & 1 & -2 & 1
\end{bmatrix}$$

which implies that

$$\boldsymbol{y} = (\boldsymbol{\lambda F} + \boldsymbol{I})\boldsymbol{x}$$

Thus, the HP trend is given by:

$$\hat{\boldsymbol{x}} = (\boldsymbol{\lambda F} + \boldsymbol{I})^{-1}\boldsymbol{y}$$

and

$$\hat{\boldsymbol{c}} = \boldsymbol{y} - \hat{\boldsymbol{x}}$$

In the figure 1.10, the λ value is 80 and the trend and cyclical information is separated from the natural log SP500 index and the trend data is useful for the long term investors like pension fund manager. The long term fund managers require the

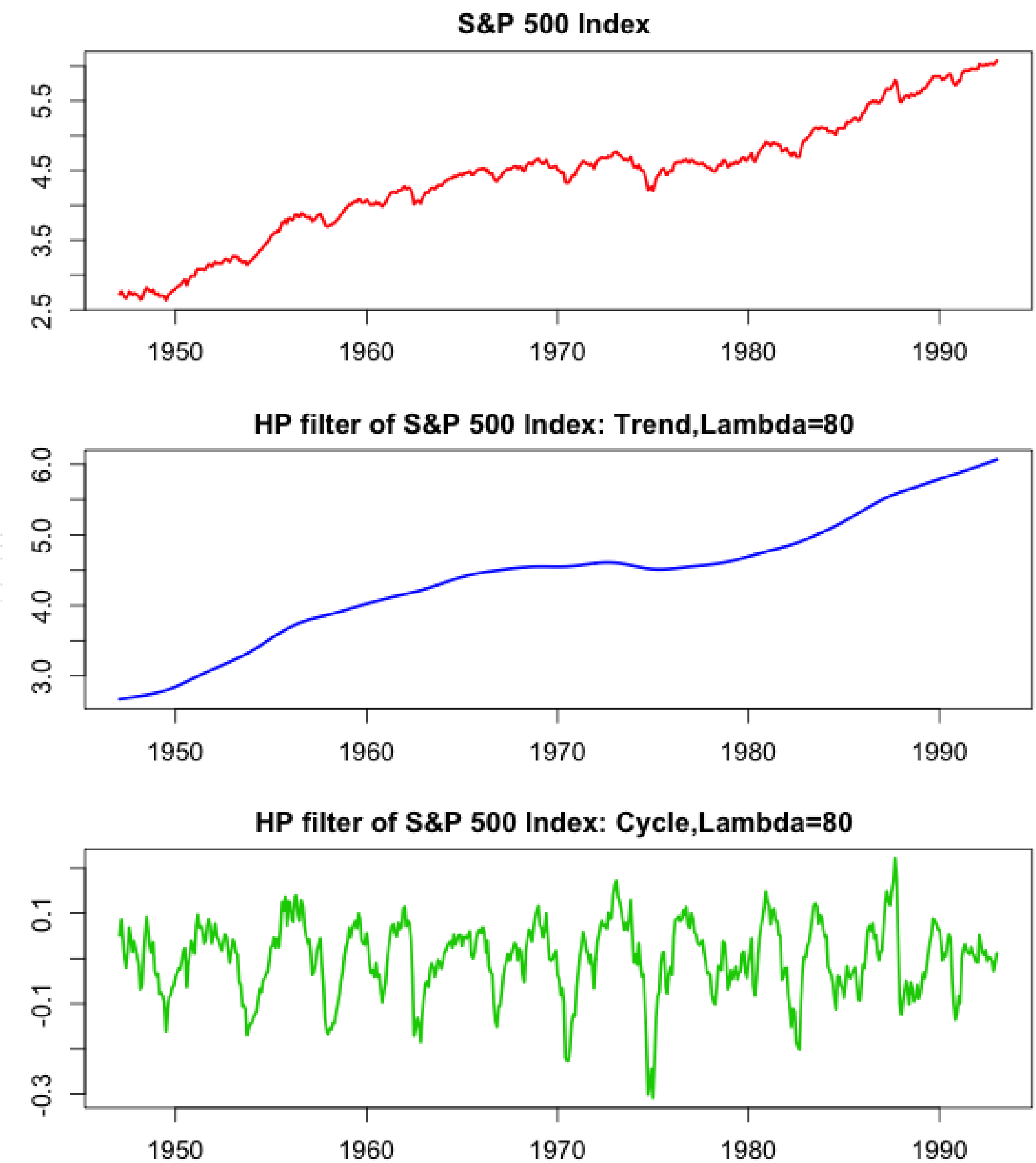


Figure 1.10: The trend and cycle separation from SP500 using HP filter with $\lambda = 80$. a) Natural Log of SP500 index b) Trend extracted from Log of SP500 index using HP filter c) Cyclical pattern extracted from SP500 index using HP filter

projection of the performance of an index to strategize the investment to meet the client's retirement objective.

In the figure 1.11, the SP500 trend analysis was performed using different λ values. The trend looks very similar when the λ is above 800 value and there is no significance

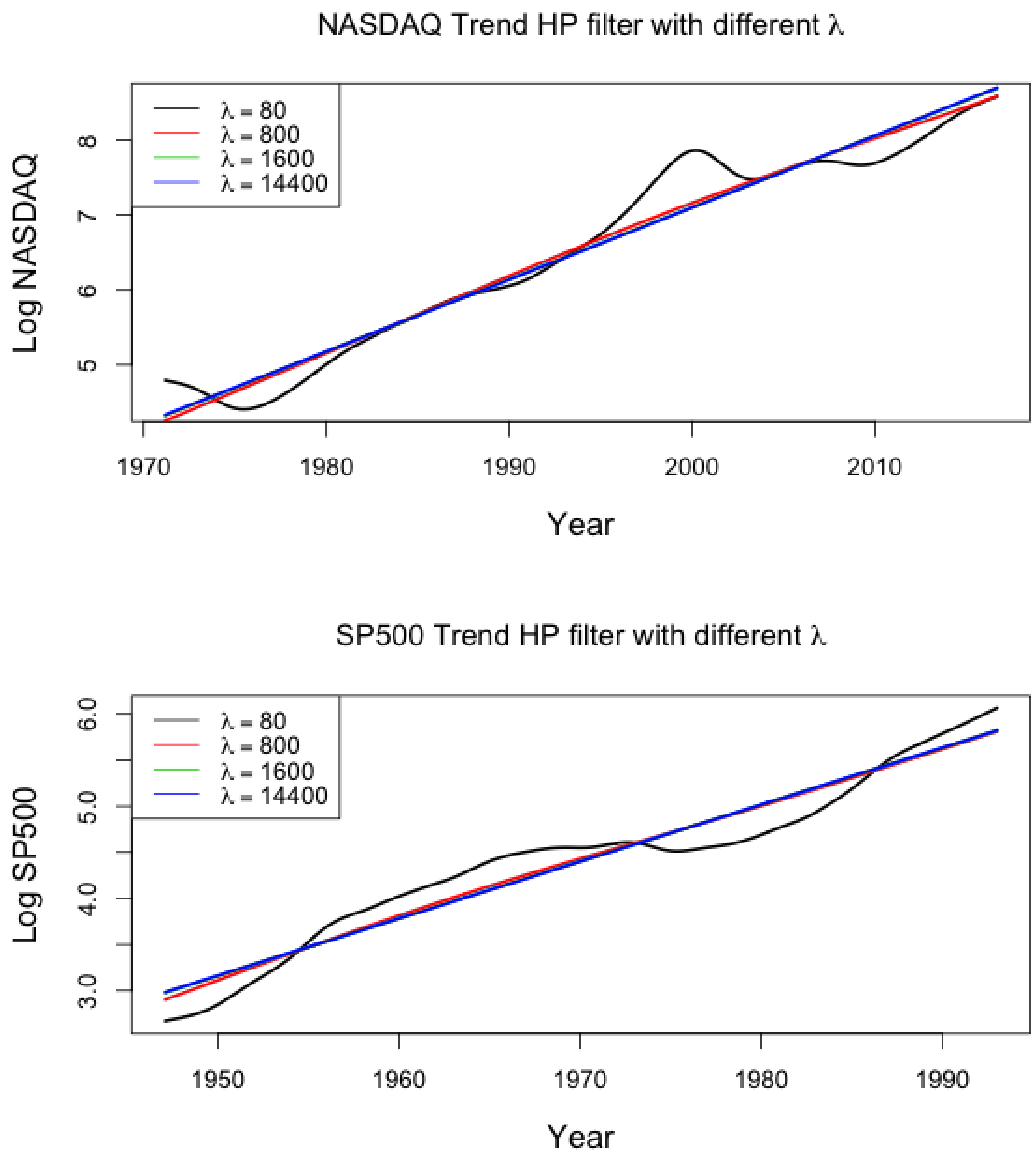


Figure 1.11: HP filter with different value for λ a) NASDAQ b) S&P 500

when the λ values are at 1600, 14400. The above figure 1.11, indicates that when λ is above 800 or above the trend in the HP filter is very close to least square linear regression line.

In the figure 1.12, the λ value is 800 and a similar analysis was performed.

In the figure 1.13, the λ value is 80 and the trend and cyclical information is separated from the natural log NASDAQ index.

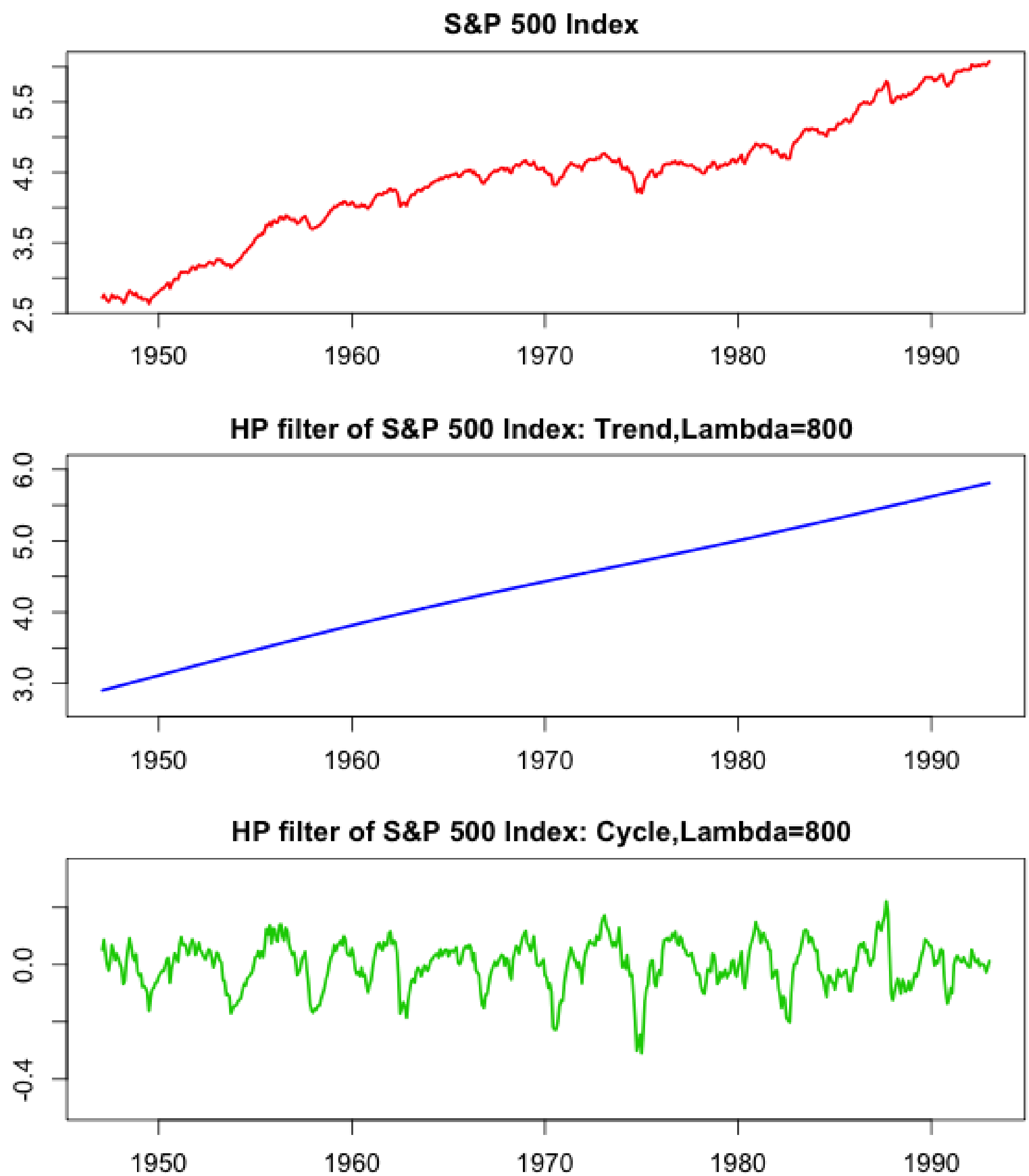


Figure 1.12: The trend and cycle separation from SP500 using HP filter with $\lambda = 800$. a) Natural Log of SP500 index b) Trend extracted from Log of SP500 index using HP filter c) Cyclical pattern extracted from SP500 index using HP filter

In the figure 1.11, the NASDAQ trend analysis was performed using different λ values. The trend looks very similar when the λ is above 800 value and there is no signficance when the λ values are at 1600, 14400.

The HP filters is one of the most heavily used econometric methods for measuring

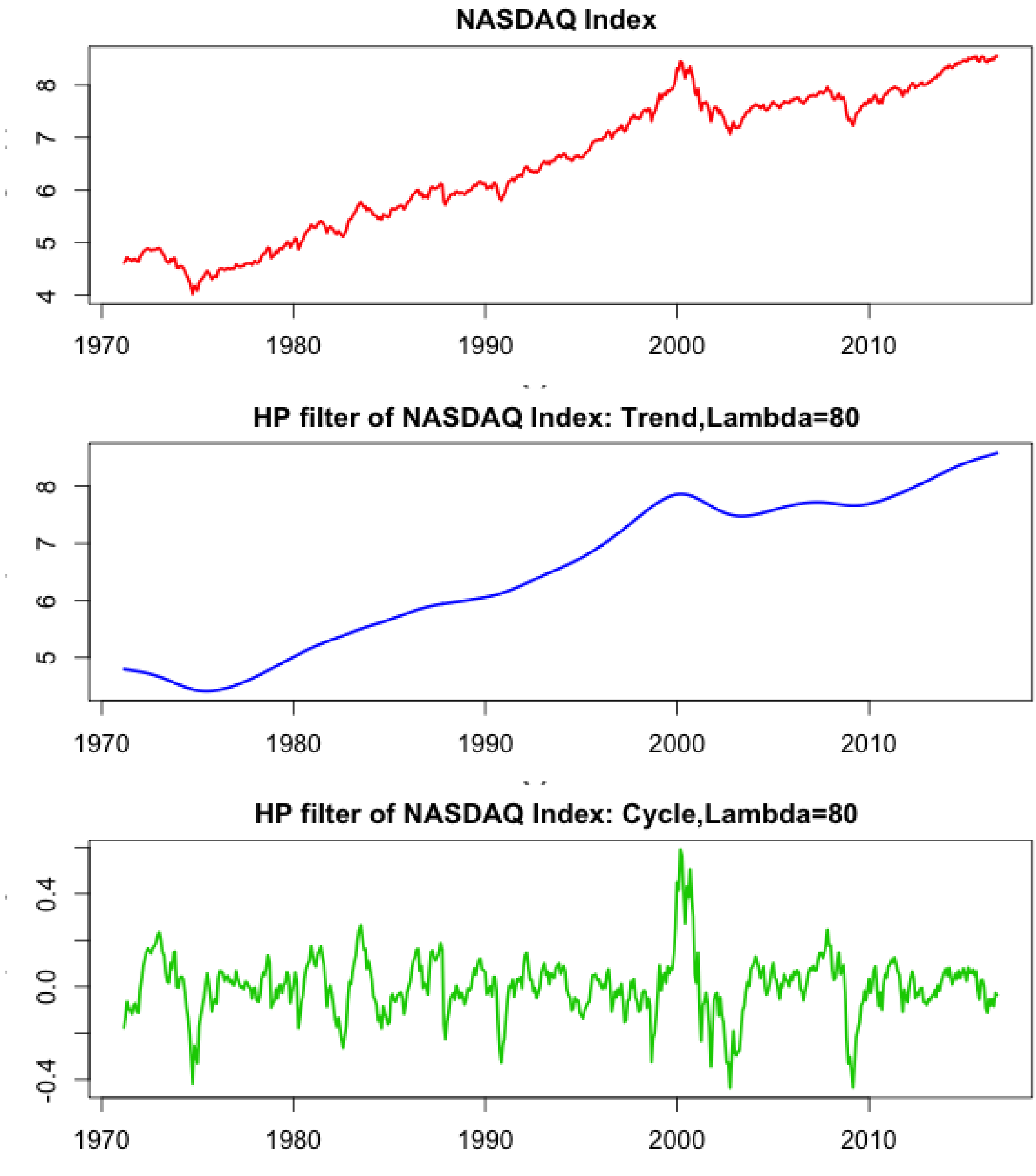


Figure 1.13: The trend and cycle separation from NASDAQ using HP filter with λ = 80. a) Natural Log of NASDAQ index b) Trend extracted from Log of NASDAQ index using HP filter c) Cyclical pattern extracted from NASDAQ index using HP filter

business cycles and potential output in empirical research. It is also a smoothing method that belongs to a very general class of nonparametric graduation procedures that depend on a tuning parameter governing the properties of the smoother. The long run potential output of an economy can be substantially influenced by great

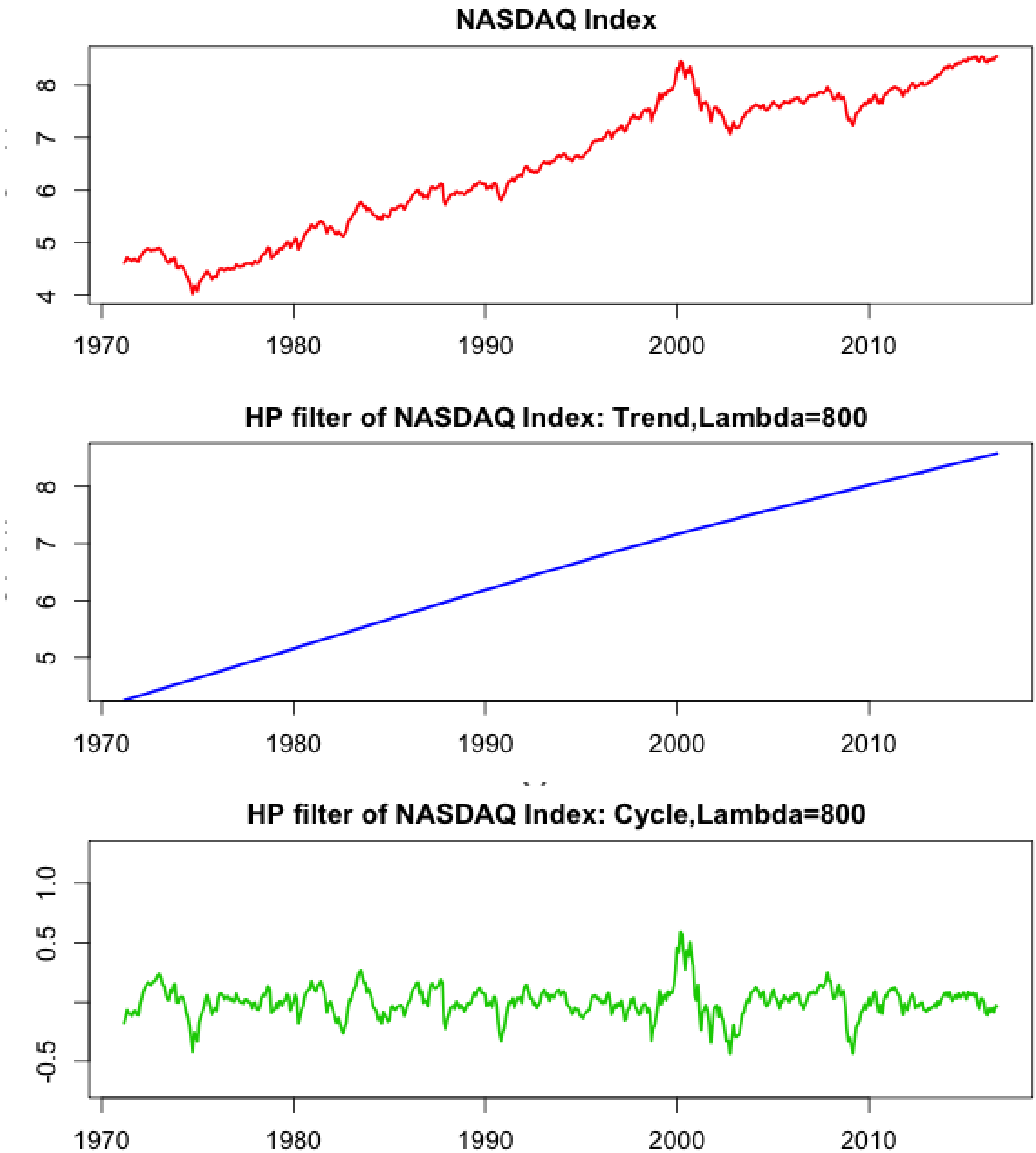


Figure 1.14: The trend and cycle separation from NASDAQ using HP filter with λ = 800. a) Natural Log of NASDAQ index b) Trend extracted from Log of NASDAQ index using HP filter c) Cyclical pattern extracted from NASDAQ index using HP filter

recessions and depressions, which may sufficiently divert resources to impact long run trend components of output. The HP filter has the advantage that, depending on the smoothing parameter (λ) choice, it can encompass long run behavior that encompasses a vast range of possibilities -from a deterministic linear trend, to a smooth Gaussian process, through to stochastic trends and combination of stochastic trends and deterministic trends that even include trend breaks.

CHAPTER II

Gabor Transformation

‘

2.1 Signal Processing

An arbitrary signal given by a function $f(t)$ can be represented by many forms for better understanding of the signal. The representation of the signal $f(t)$ in a different form depends on type of application the user is interested in. In few cases, the original signals are perfectly fine as is for a given application.

In signal processing there are two major field of study.

- Signal synthesis - Construction of a signal
- Signal analysis - Study of a signal

Gabor in his original paper - Theory of communication [**?**], stated that a time function $f(t)$ in the time interval $t_2 - t_1 = \tau$ contains an infinite data and it can be signal in τ can be further sub divided into N sub-intervals, and define, for instance, the average ordinate in each sub-interval as a "datum" . There are infinite ways to represent $f(t)$ in the internal τ and one of the way is to represent $f(t)$ in the interval τ by a polynomial of order N, to fit it as closely as possible of $f(t)$ by the method of least squares, and to take the coefficients of the polynomial as data. . The polynomial coefficients of the curve is the data represents the curve in the interval. The polynomial can be specified such a way that moments M_n equal to as follows and it is equivalent of the polynomial coefficient

$$M_0 = \int_0^\tau f(t)dt; M_1 = \int_0^\tau tf(t)dt; M_2 = \int_0^\tau t^2 f(t)dt; ..M_{N-1} = \int_0^\tau t^{N-1} f(t)dt$$

The function $f(t)$ in the interval τ is expanded in the terms of set of powers of time functions. If the purpose is to transmit the signal,then the moments (equivalent of polynomial coefficients) can be transmitted and the signal can be reconstructed at the other end.

Instead of representing the function $f(t)$ in the interval τ in terms of powers of time functions, it can be represented by orthogonal functions $\phi_k(t)$ in the interval 0 $<$ t $< \tau$ and it is equivalent of fitting the expansion. How close the fit will be depends on the set of orthogonal functions selected and type of applications.

2.2 Fourier Transformation

If the orthogonal function set is simple harmonic functions sine and cosine in the interval extending from $-\infty$ to ∞, then the given signal is presented in the frequency domain and it is called the Fourier transformation.

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ts}dt; \tag{2.1}$$

$F(s)$ is the Fourier transform of $f(t)$. The inverse of the $F(s)$ provides the function $f(t)$ and the equation is given below.

$$f(t) = \int_{-\infty}^{\infty} F(s)e^{j2\pi ts}ds \tag{2.2}$$

Fourier transformation is a tool to translate a signal from time domain to frequency domain and vice versa. In fact, it is an apt tool to analyze a signal in the frequency domain given the signal does not evolve over the time, but, most of the practical signals like music, seismic and others evolve over the time. It is challenging to study and understand the insights of those signals in the frequency domain by just using Fourier transformation. In recent years, there are many ideas proposed to over come that challenge and gain more insights of the signals in other domains.

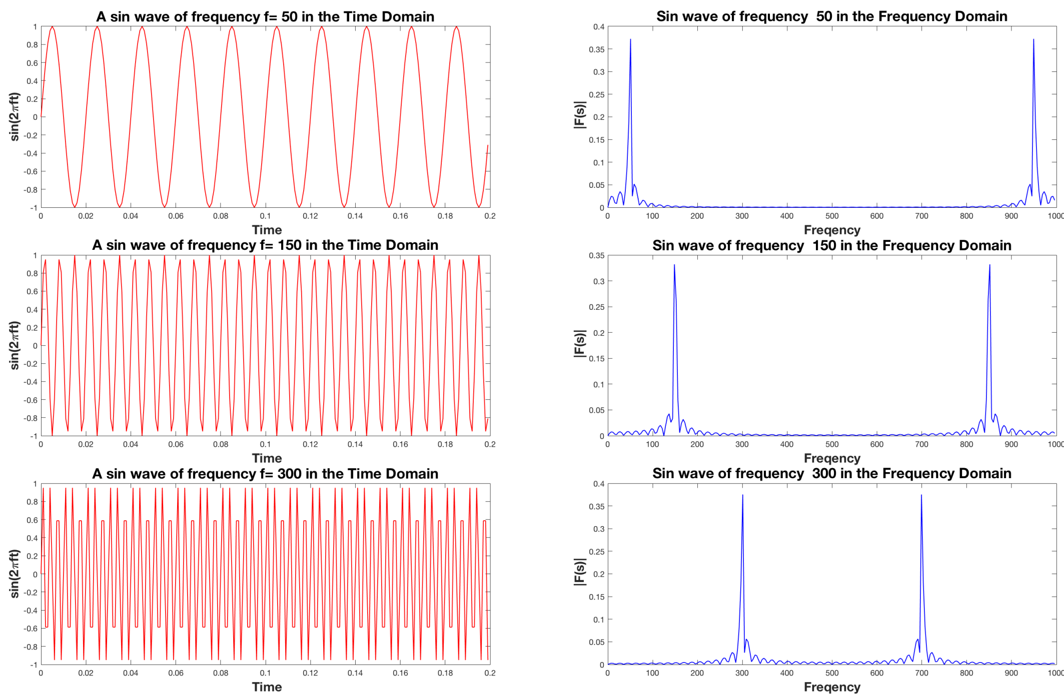


Figure 2.1: Fourier Transform for a discretize time signal sin wave with three different frequencies and respective discrete Fourier transform is given above. The graph was created using Matlab code DrawSinFourierGraph.m and it is attached in the Appendix B

In the above figure **??**, a sin wave with different frequencies $\omega = 50, 150, 300$ is transformed into a frequency domain using the Fourier transform. Please note high amplitude in the frequency domain for their respective frequency in each graph.

$$f(t) = \sin(2\pi\omega t); \tag{2.3}$$

Three type of sin wave created with frequencies $\omega = 50, 150, 300$.

The Fourier transform of the $f(t)$ is given by

$$F(s) = \int_{-\infty}^{\infty} \sin(2\pi\omega t) e^{-j2\pi t\omega} dt; \tag{2.4}$$

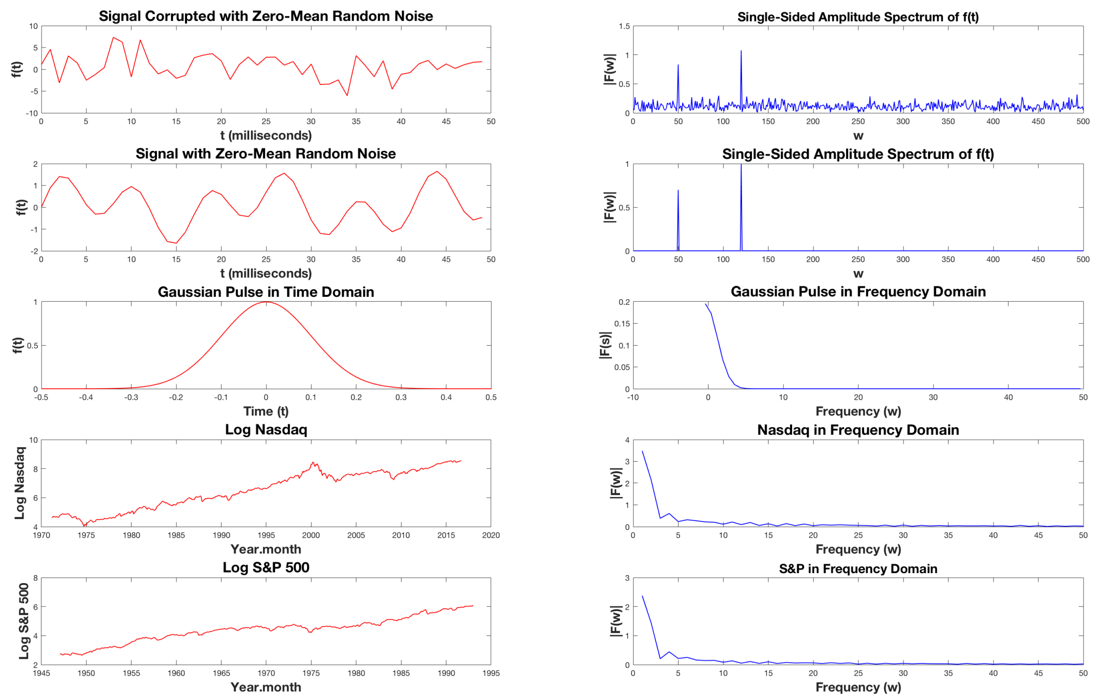


Figure 2.2: Fourier transform for a) Zero mean signal with random noise, b) Zero mean signal, c) Gaussian , d) Log NASDAQ, e) Log S&P 500

In the above figure 2.2 , a signal with zero mean random noise, signal with zero mean with no random noise, Gaussian curve, *log* sp500, *log* Nasdaq are also transformed into a frequency domain using the Fourier transform. F(ω) is an even function and so only $f > 0$ is shown in the 2.2. As shown in the above 2.2, in the frequency domain, F(ω) does not provide when (time - t) the respective frequency occurred.

2.2.1 Short Time Fourier Transformation

One of the ideas is to chop the signal into smaller pieces and perform Fourier transformation for each piece. This technique is called short time Fourier Transform. The smaller pieces in the signal can be chosen by a window function $w(t-\tau)$

$$F(\omega,\tau)=\int_{-\infty}^{\infty} f(t)w(t-\tau)e^{-j2\pi t\omega}dt; \tag{2.5}$$

$w(t)$ represents a window function and there are multiple window functions available to choose.

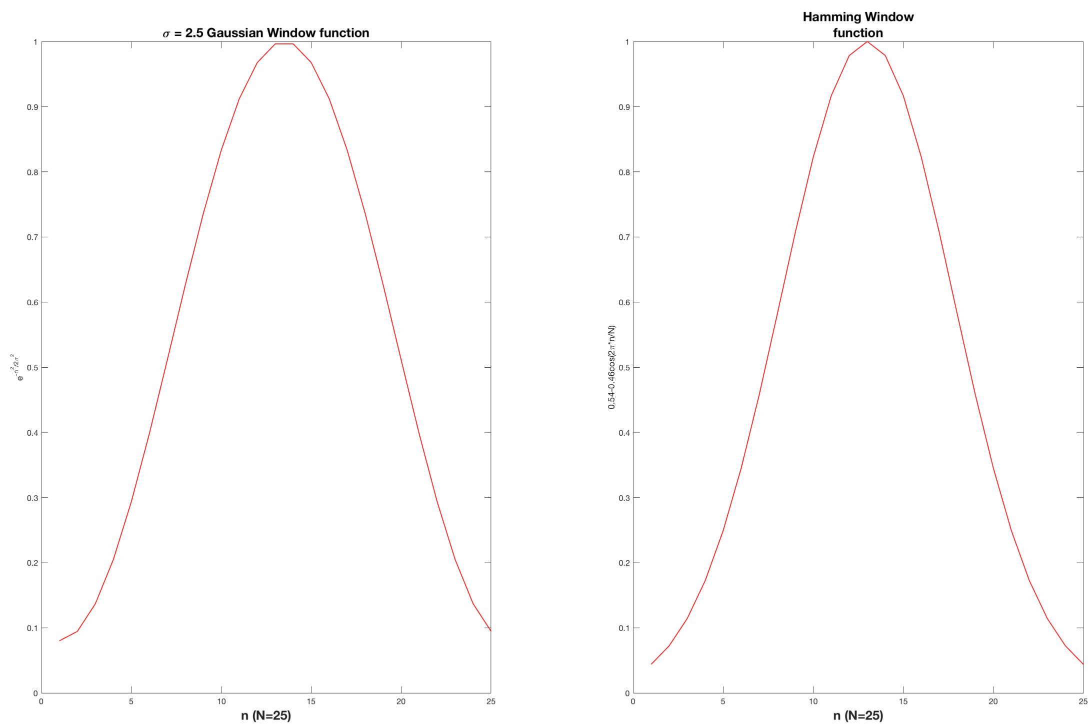


Figure 2.3: Window function (window length =25) for a) Gaussian b) Hamming. The graph was created using Matlab program ghamwin.m and attached in Appendix B

The short time Fourier transform is performed using Hamming and Gaussian window and its results are given below.

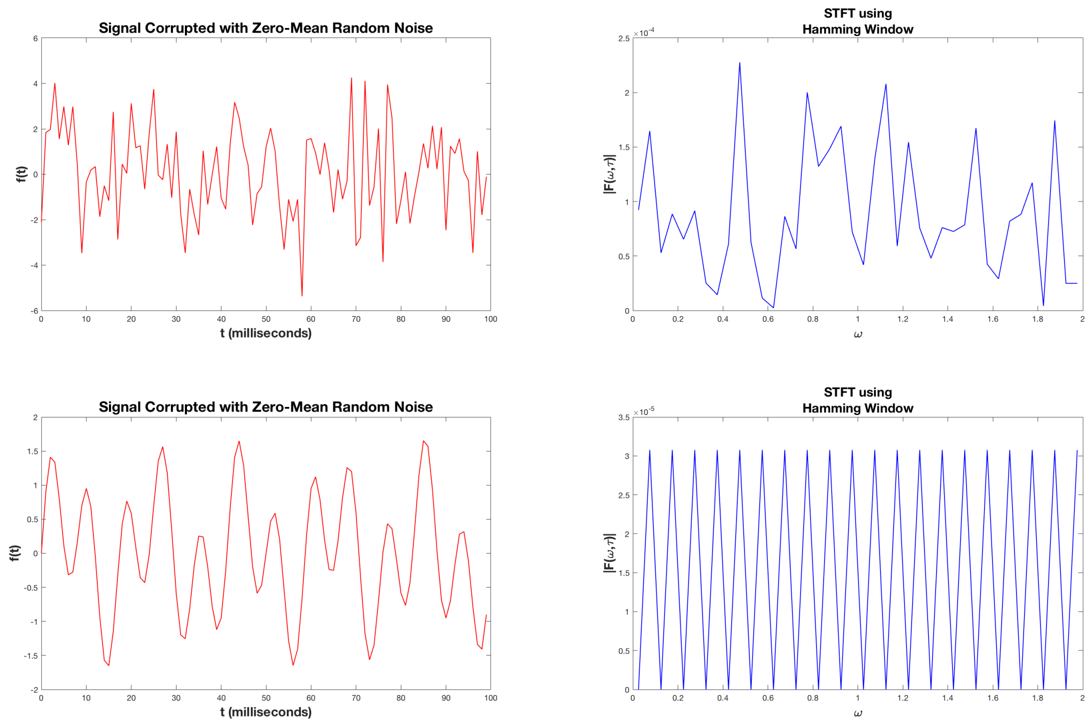


Figure 2.4: Window Length=25; τ = 25 Hamming Window function used for short time Fourier Transform for a) Zero Mean Signal with Random Noise b) Zero Mean signal. The graph was created using Matlab program drawSTFTEg.m and attached in Appendix B

Simultaneous analysis of signals in both time and frequency domains provides better understanding of the signal and short time Fourier transform laid the foundation for the joint time-frequency analysis.

2.3 Introduction

Gabor transformation or expansion in signal synthesis uses the Gabor elementary function (GEF) as the base function (equivalent of simple harmonic function in the Fourier transform) and idea was influenced by Heisenberg's uncertainty principle.

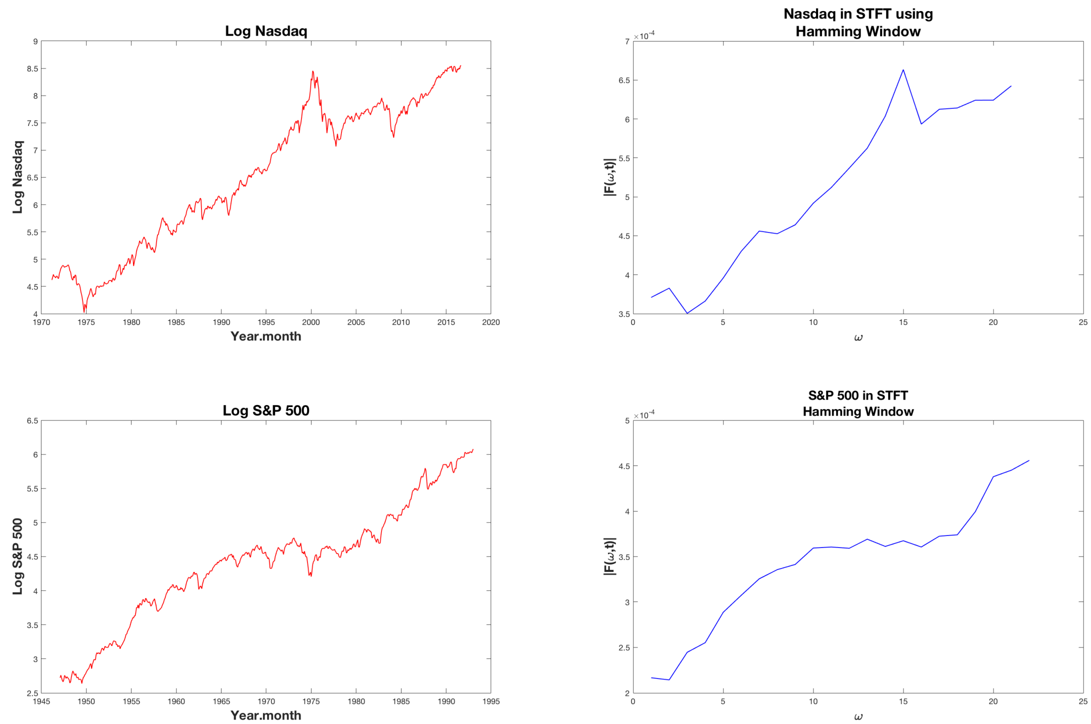


Figure 2.5: Window Length=25; $\tau = 25$ Hamming Window function used for short time Fourier Transform for a) Log NASDAQ b) Log SP500. The graph was created using Matlab program drawSTFTEco.m and attached in the Appendix B

2.3.1 Heisenberg's uncertainty principle

In quantum mechanics, simultaneously, both position of a particle and momentum of the particle can not be measured preciously. Let x be position and p be momentum of the particle. Standard deviation of x and p are given by Δx and Δp respectively. Uncertainty principle states that product of variance of position and momentum is greater than or equal to $\frac{\hbar}{2}$

$$\Delta x \Delta p \geq \frac{\hbar}{2}$$

$$\Delta x = \sqrt{\langle (x - \langle x \rangle)^2 \rangle}$$

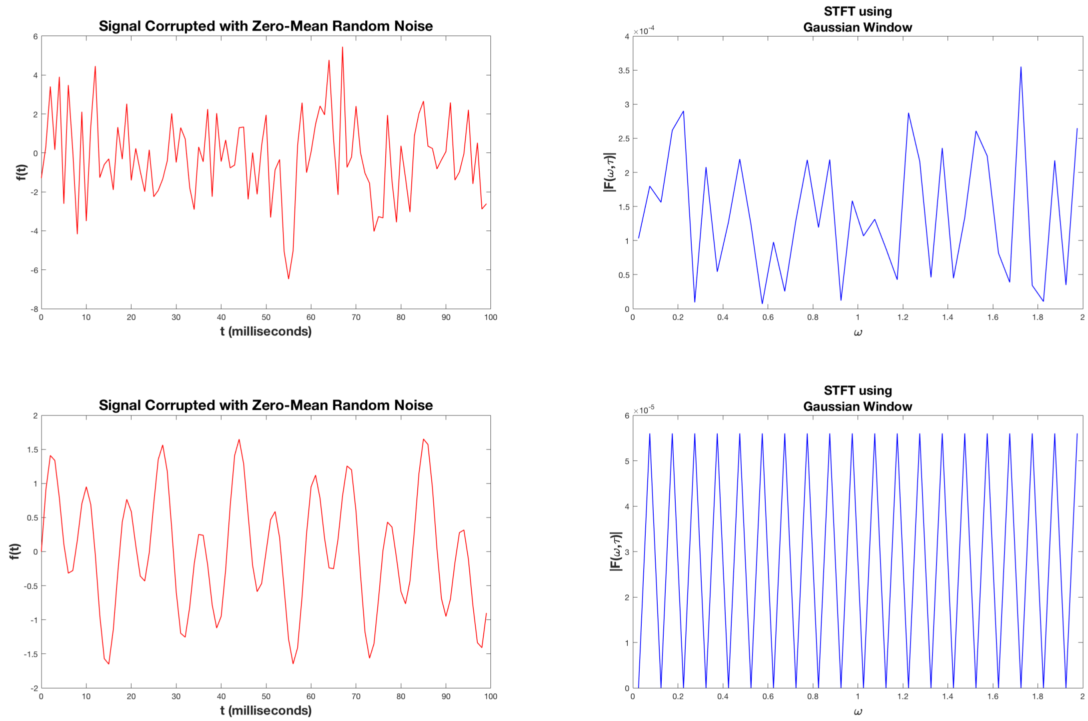


Figure 2.6: Window Length=25; $\tau = 25$ Gaussian Window function used for short time Fourier Transform for a) Zero Mean Signal with Random Noise b) Zero Mean signal. The graph was created using Matlab program drawSTFTEco.m and attached in the Appendix B

$$\Delta p = \sqrt{\langle (p - \langle p \rangle)^2 \rangle}$$

2.3.2 Gabor Transformation

Gabor had the insight the base function with minimum uncertainty in both time and frequency domains best captures temporal information during the frequency analysis.

Gabor in his original study proposed an elementary functions in the complex form which occupies minimum uncertainty and it is a product of harmonic oscillator of any frequency and probability function. The area occupied by elementary function in the

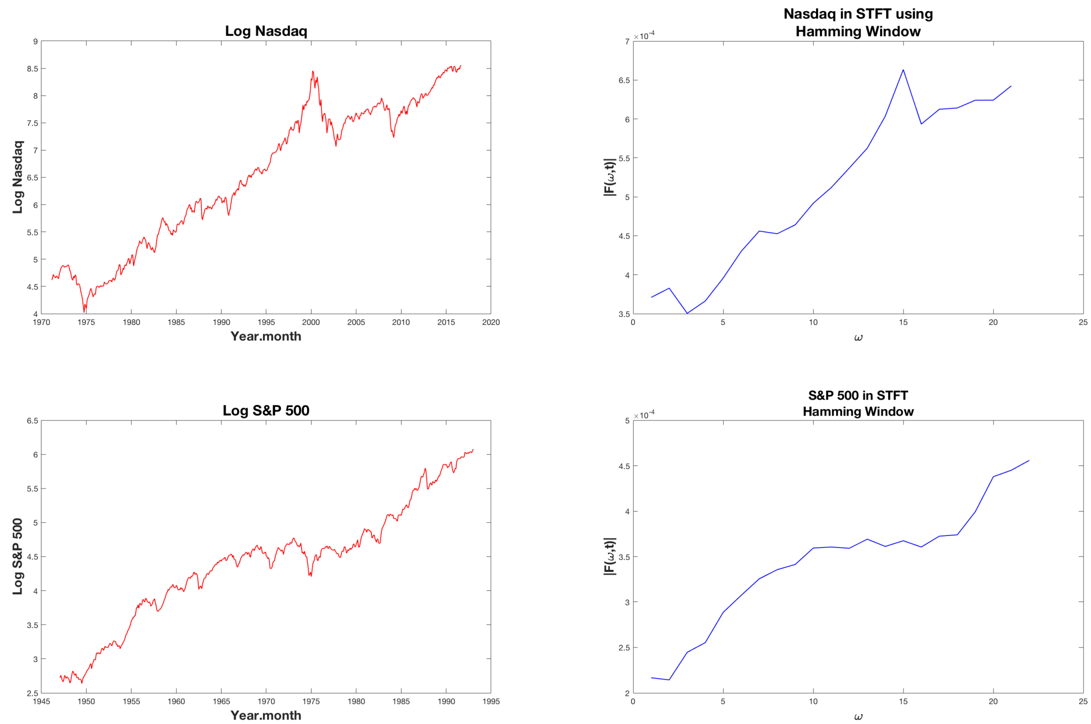


Figure 2.7: Window Length=25; $\tau = 25$ Gaussian Window function used for short time Fourier Transform for a) Log NASDAQ b) Log SP500.The graph was created using Matlab program drawSTFTEco.m and attached in the Appendix B

joint time frequency domain is equal to minimum uncertainty.

$$\psi(t) = \underbrace{e^{-\alpha^2(t-t_0)^2}}_{v} \overbrace{e^{j2\pi f_0 t+\phi}}^{w} \tag{2.6}$$

An arbitrary function $f(t)$ can be represented by series of elementary functions which are constructed by translation in both time and frequency domains. The function $f(t)$ is synthesized by the combination of GEF.

$$f(t) = \sum_{m,n \in Z} C_{m,n}\psi_{m,n}(t) \tag{2.7}$$

where the $C_{m,n}$ are Gabor coefficients and the $\psi_{m,n}(t)$ are the bases functions called Gabor elementary functions. GEF shifted or translated by 'a' and generation of the translation of $\psi_{m,n}$ by 'na'is given by

$$\psi_n(t) = \psi(t - na)$$

GEF shifted or translated and modulated (translation in the frequency domain is also called modulation) by the simple harmonic functions.

$$\psi_{m,n}(t) = \psi(t - na)e^{j2\pi mbt}$$

The original Gabor paper suggested that ψ is Gaussian. Later GEF was studied by using other functions for $\psi_{m,n}$ like a rectangle. a,b are time and frequency shift parameters and a,b > 0. $\psi_{m,n}$ is obtained by shifting it by lattice na $\times$ mb in time-frequency plane.

Gabor co-efficient $C_{m,n}$ and synthesis function $f(t)$ are bi-orthogonal functional sets.

$$C_{m,n} = \sum_{m,n \in Z} \psi^*_{m,n}(t) f(t) \tag{2.8}$$

$\psi^*_{m,n}$ is the complex conjugate of $\psi_{m,n}$.

GEFs could be a set of Gaussian functions modulated by simple harmonics. GEF generated in 1D by combining Gaussian and simple harmonic functions are given in the figure 2. In the figure 2, Gaussian function remained the same for both GEF and only the sinusoidal functions changed between the two GEFs. Note the change in the

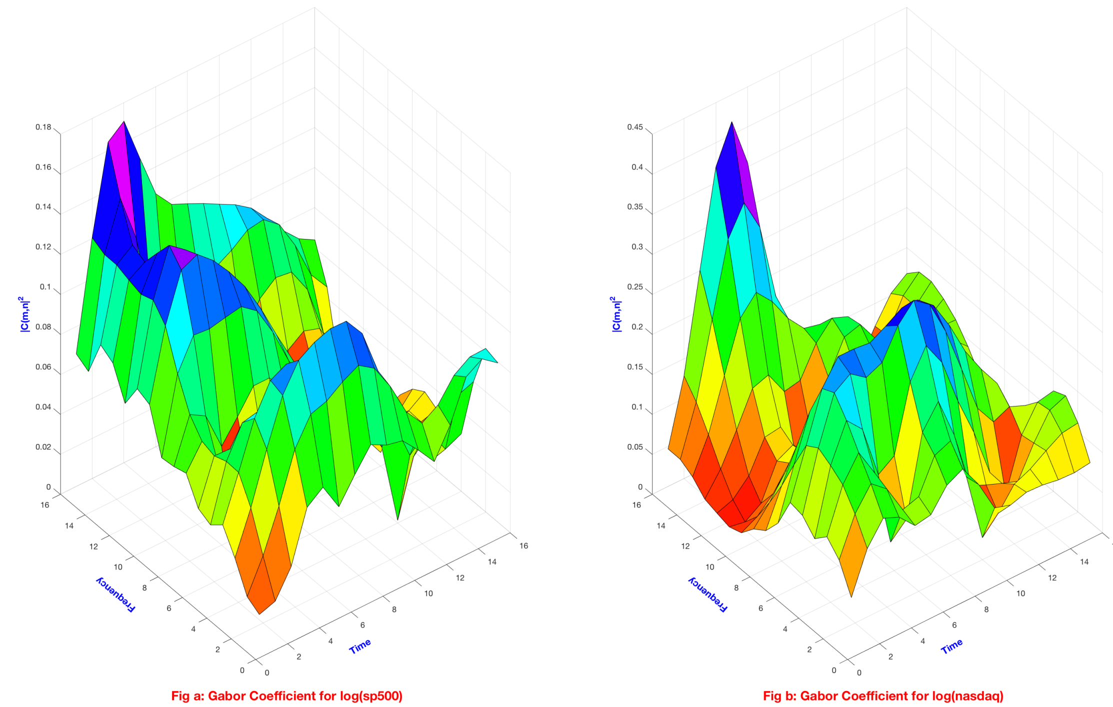
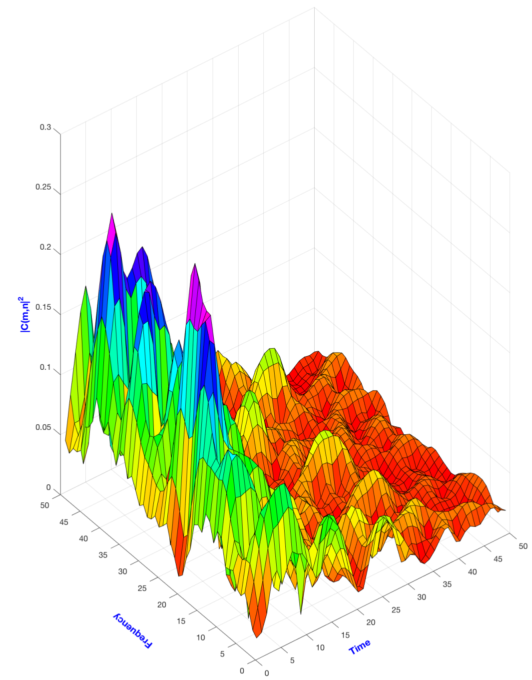


Figure 2.8: Fig a is the Gabor Coefficient for HP Filter cycles of log(SP500) and Fig b is the Gabor Coefficient for HP Filter cycles of log(NASDAQ).$\Delta M = 8; \Delta N = 4; m = 16; n = 16; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$ The program used to create the graph is mygabor.m and it is attached in the appendix.

frequency between the GEFs. A set of GEFs can be created by varying na and mb in equ(3).

$$\psi_{m,n}(t) = e^{-\frac{(t-\mu-na)^2}{2\sigma^2}} e^{j2\pi mbt}$$

If GEF ψ and its Fourier transform (Frequency representation) Ψ are localized at the origin and $g_{m,n}$ localized at (na,mb) in the joint time-frequency domain. Each GEF occupies a region in time-frequency plane and associated $C_{m,n}$ represents quantum of information.



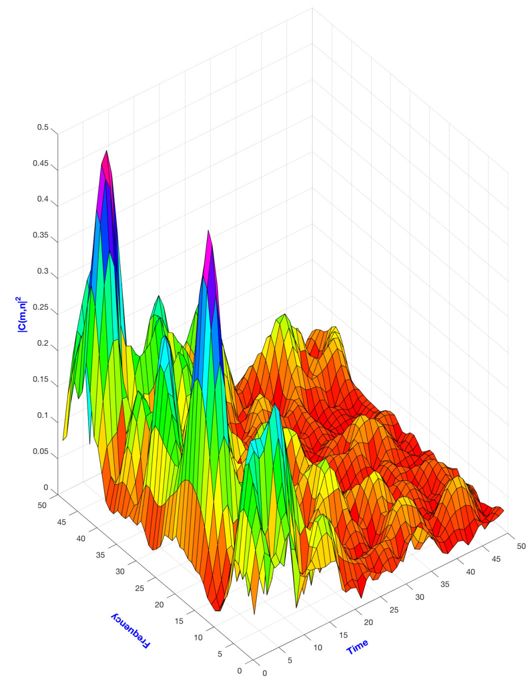


Figure 2.9: Fig a is the Gabor Coefficient for HP Filter cycles log(SP500) and Fig b is the Gabor Coefficient for HP Filter cycles log(NASDAQ).$\Delta M = 8; \Delta N = 4; m = 50; n = 50; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$ The program used to create the graph is mygabor.m and it is attached in the appendix.

Let $\psi(t)$ be GEF and GEF in the frequency domain is given by Fourier transform

$$\Psi(f) = \int_{-\infty}^{\infty} \psi(t) e^{-j2\pi ft} \, dt$$

GEF are set of Gaussian functions modulated by simple harmonics. These functions has a special property that adheres to Heisenberg's uncertainty principle. The product of the variance in time Δt and variance in frequency Δf is always greater than or equal to a certain quantity.

$$\Delta f \Delta t \geq \frac{1}{4\pi} \tag{2.9}$$

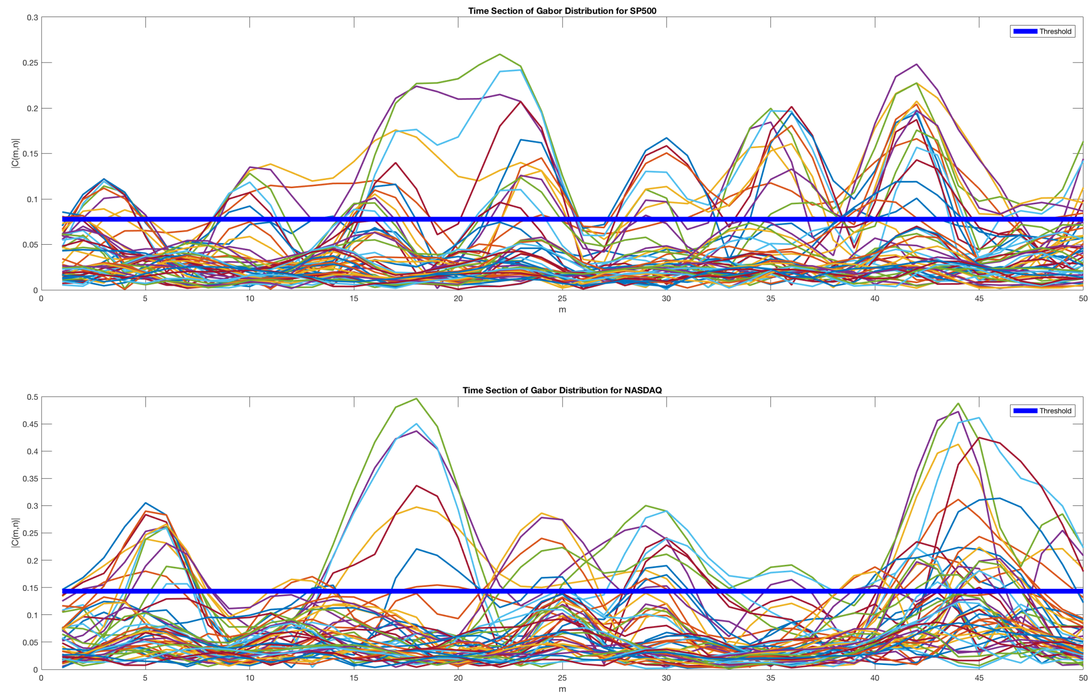


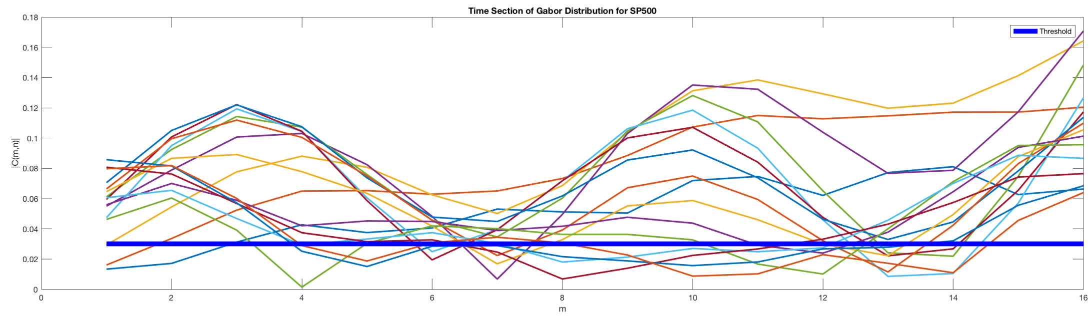
Figure 2.10: Fig a is the time section of Gabor distribution of HP Filter cycles log(SP500) and Fig b is the time section of Gabor distribution of for HP Filter cycles log(NASDAQ). $\Delta M = 8; \Delta N = 4; m = 50; n = 50; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. Threshold is calculated as $max(|c(m,n)| - min(|c(m,n)|))/2$. The program used to create the graph is mygabor.m and it is attached in the appendix.

The time variance or effective duration and frequency variance or effective frequency width can be calculated by the root mean square (RMS) deviation of the signal from the mean. The effective duration (Δt) and effective frequency width (Δf) are given by

$$\Delta t = \sqrt{\frac{\int_{-\infty}^{\infty} \psi(t)(\mu_t - t)^2 \psi^*(t) dt}{\int_{-\infty}^{\infty} \psi(t)\psi^*(t) dt}}; \Delta f = \sqrt{\frac{\int_{-\infty}^{\infty} \Psi(f)(\mu_f - f)^2 \Psi^*(f) df}{\int_{-\infty}^{\infty} \Psi(f)\Psi^*(f) df}}$$

where μ_t and μ_f are mean time and mean frequency and it is given by

$$\mu_t = \frac{\int_{-\infty}^{\infty} \psi(t) t \psi^*(t)}{\int_{-\infty}^{\infty} \psi(t)\psi^*(t)}; \mu_f = \frac{\int_{-\infty}^{\infty} \Psi(f) f \Psi^*(f)}{\int_{-\infty}^{\infty} \Psi(f)\Psi^*(f)}$$



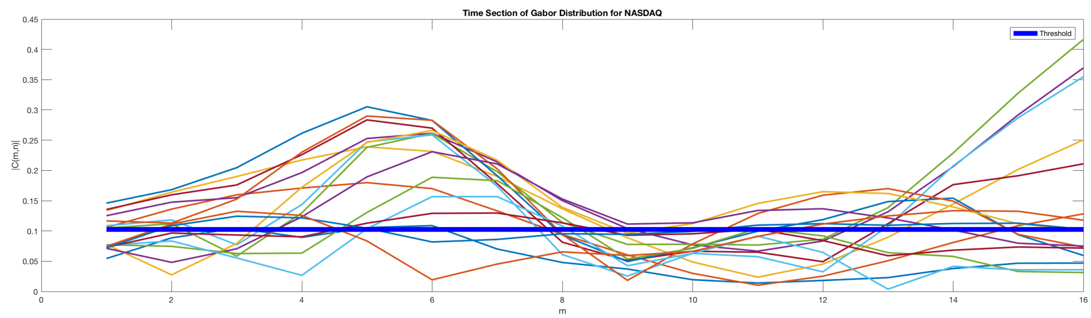


Figure 2.11: Fig a is the time section of Gabor distribution of HP Filter cycles log(SP500) and Fig b is the time section of Gabor distribution of for HP Filter cycles log(NASDAQ). $\Delta M = 8; \Delta N = 4; m = 16; n = 16; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. Threshold is calculated as $max(|c(m,n)| - min(|c(m,n)|))/2$. The program used to create the graph is mygabor.m and it is attached in the appendix.

$$\Delta t \Delta f = \sqrt{\frac{\int_{-\infty}^{\infty} \psi(t)(\mu_t - t)^2 \psi^*(t)dt}{\int_{-\infty}^{\infty} \psi(t)\psi^*(t)dt} \frac{\int_{-\infty}^{\infty} \Psi(f)(\mu_f - f)^2 \Psi^*(f)df}{\int_{-\infty}^{\infty} \Psi(f)\Psi^*(f)df}} \geq \frac{1}{4\pi}$$

There are three possibilities for above equation is given below in table 2.1 .

$\Delta t\ \Delta f$	**S**ampling	Remarks
$= \frac{1}{4\pi}$	Critical	Special functions called GEF
$> \frac{1}{4\pi}$	Over	–
$< \frac{1}{4\pi}$	Under	–

Table 2.1: Possible values for Δt Δf and special case for Gabor function

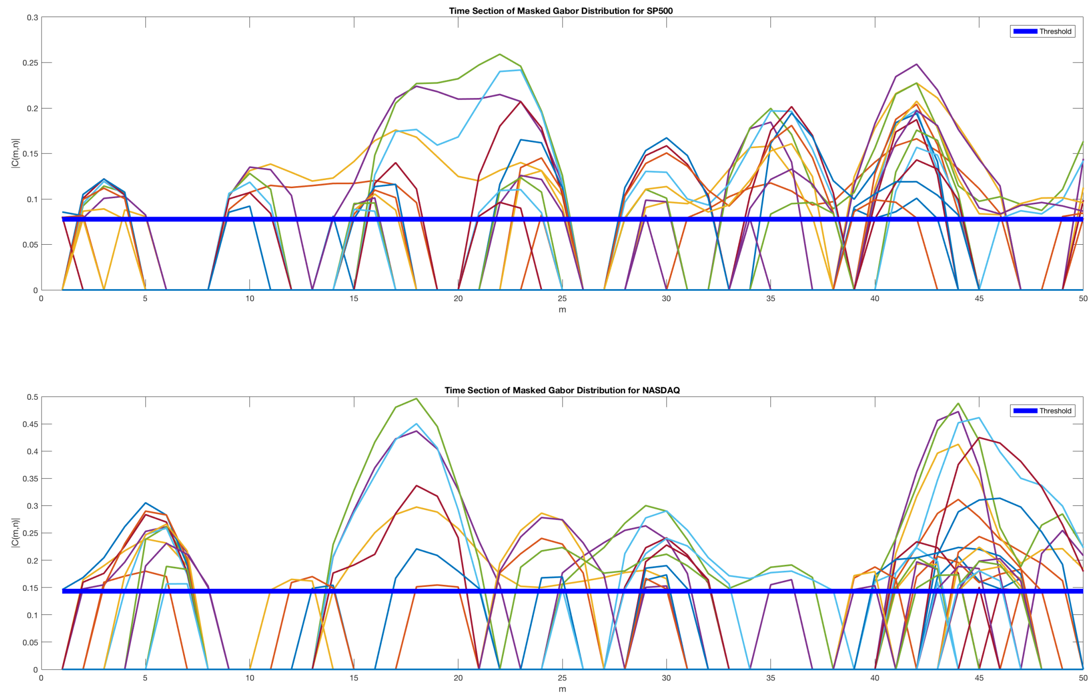
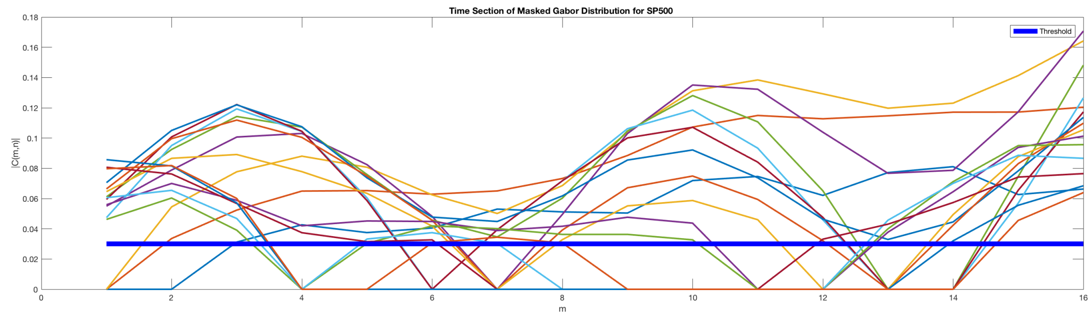


Figure 2.12: Fig a is the time section of Gabor distribution of HP Filter cycles log(SP500) and Fig b is the time section of Gabor distribution of for HP Filter cycles log(NASDAQ). $\Delta M = 8; \Delta N = 4; m = 50; n = 50; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. Mask Operator M created based on threshold. The program used to create the graph is mygabor.m and it is attached in the appendix.

Analogous to 1D uncertainty principle, there are two 2D uncertainty principles constraining the effective width (Δx) and the effective length (Δy) of a signal $f(x, y)$ and the effective width (Δu) and the effective length (Δv) of its 2D Fourier transform $F(u, v)$

$$\Delta x \Delta u = \sqrt{\frac{\int_{-\infty}^{\infty} \psi(x,y)(\mu_x - x)^2 \psi^*(x,y) dxdy}{\int_{-\infty}^{\infty} \psi(x,y)\psi^*(x,y) dxdy} \frac{\int_{-\infty}^{\infty} \Psi(u,v)(\mu_u - u)^2 \Psi^*(u,v) dudv}{\int_{-\infty}^{\infty} \Psi(u,v)\Psi^*(u,v) dudv}} \geq \frac{1}{4\pi}$$



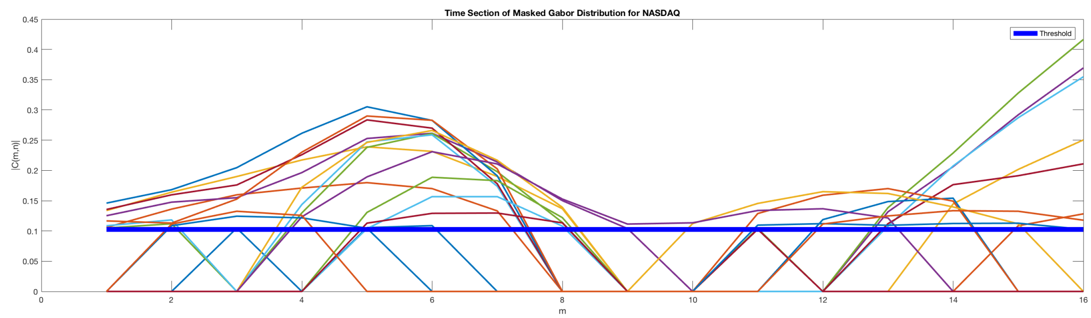
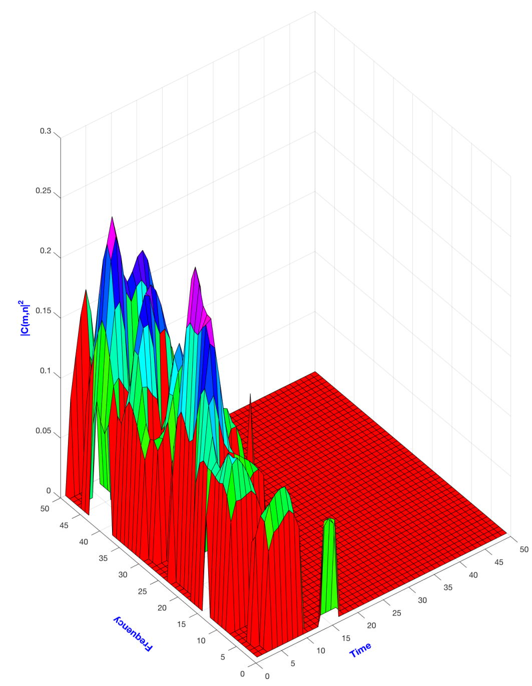


Figure 2.13: Fig a is the time section of Gabor distribution of HP Filter cycles log(SP500) and Fig b is the time section of Gabor distribution of for HP Filter cycles log(NASDAQ). $\Delta M = 8; \Delta N = 4; m = 16; n = 16; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. Mask Operator M created based on threshold. The program used to create the graph is mygabor.m and it is attached in the appendix.

$$\Delta y \Delta v = \sqrt{\frac{\int_{-\infty}^{\infty} \psi(x,y)(\mu_y - y)^2 \psi^*(x,y)dxdy}{\int_{-\infty}^{\infty} \psi(x,y)\psi^*(x,y)dxdy} \frac{\int_{-\infty}^{\infty} \Psi(u,v)(\mu_v - v)^2 \Psi^*(u,v)dudv}{\int_{-\infty}^{\infty} \Psi(u,v)\Psi^*(u,v)dudv}} \geq \frac{1}{4\pi}$$

$$\Delta x \Delta u \Delta y \Delta v \geq \frac{1}{16\pi^2}$$



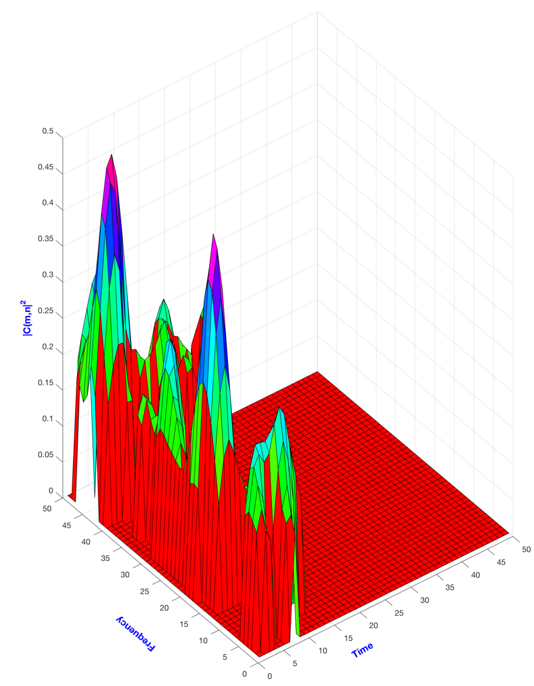


Figure 2.14: Fig a is the Filtered Gabor Coefficient for HP Filter cycles log(SP500) and Fig b is the Filtered Gabor Coefficient for HP Filter cycles log(NASDAQ).$\Delta M = 8; \Delta N = 4; m = 50; n = 50; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. The mask operator is created based on the threshold of a peak distribution. The program used to create the graph is myfiltgabor.m and it is attached in the appendix.

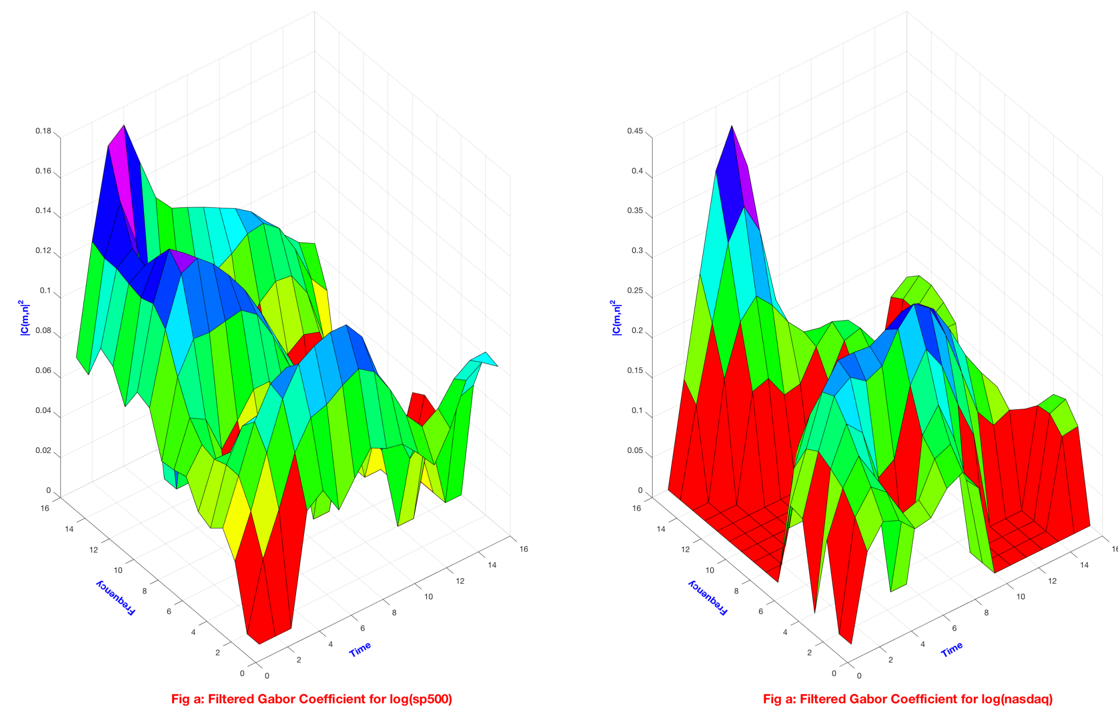


Figure 2.15: Fig a is the Filtered Gabor Coefficient for HP Filter cycles log(SP500) and Fig b is the Filtered Gabor Coefficient for HP Filter cycles log(NASDAQ).$\Delta M = 8; \Delta N = 4; m = 16; n = 16\sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. The mask operator is created based on the threshold of a peak distribution. The program used to create the graph is myfiltgabor.m and it is attached in the appendix.

CHAPTER III

Wigner Distribution

3.1 Wigner Distribution

The Wigner Distribution (WD) introduced by Wigner (Wigner 1932) as a phase representation in Quantum Mechanics gives a simultaneous representation of a signal in space an spatial-frequency variables. WD belongs to a large class of bilinear distribution known as the Cohen's class, in which each member can be obtained choosing different kernels in the generalized bilinear distribution definition.

Let suppose $f(t)$ is continuous, integrable and complex function, the symmetric definition of the Wigner distribution $W_f(t,\omega)$ is given by

$$W_f(t,\omega) = \int_{-\infty}^{\infty} f(t+\frac{\tau}{2})f^*(t-\frac{\tau}{2})e^{-j\omega\tau}d\tau \tag{3.1}$$

where t and τ are spatial variables, ω is the spatial frequency variable and f^* is the complex conjugate of f. The product function $r_f(t,\tau)$ is given by

$$r_f(t,\omega) = f(t+\frac{\tau}{2})f^*(t-\frac{\tau}{2}) \tag{3.2}$$

The auto-Wigner distribution gives a generalized auto convolution at non-zero frequency. From $W_f(t,\omega)$, it can be observed that the Wigner Distribution is the

Fourier transformation, for a given point τ, of the product ff^*. It may also be obtained from the Fourier transform, F of f by

$$W_F(\omega, t) = \int_{-\infty}^{\infty} F(\omega + \frac{\phi}{2}) F^*(\omega - \frac{\phi}{2}) e^{j\phi t} d\phi \tag{3.3}$$

where F^* is the complex conjugate of F.

According to $W_f(t, \omega)$ and $W_F(\omega, t)$ the following relation is observed,

$$W_f(t, \omega) = W_F(\omega, t) \tag{3.4}$$

which shows the symmetry between the two conjugate domains. Among various properties of WD, the interference property is more relevant for the current study. The WD computation introduces spurious "auto term" due to its intrinsic bi-linearity. The WD of sum of two signals $f(t) + f'(t)$ is given by

$$W_{f+f'}(t, \omega) = W_f(t, \omega) + W_{f'}(t, \omega) + 2Re[W_{f,f'}(t, \omega)] \tag{3.5}$$

3.2 Discrete Wigner Distribution

One of the main disadvantages of the discrete definition is that not all the properties of the continues WD are preserved by discretization due to aliasing effects. Several alternatives definitions have been proposed in the literature in order to overcome this problem (Chan 1982), (Claasen 1983), (Brenner 1983), (Day 1983), (Peyrin 1986). The discrete WD of a sampled function $f(t)$ is defined by

$$W_f(t, \omega) = 2 \sum_{k=0}^{N-1} f(n+k) f^*(n-k) e^{-2j\omega k} \tag{3.6}$$

where n and $\omega = 2\pi m/N$ are the spatial and frequency variables respectively.

The discrete WD definition given above retain the basic properties of the continuous WD, however, in the main differences comes from the inversion property. In the case of discrete signals, inserting $k = n$ in the above equation allows one to write

$$f(2n)f^*(0) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} W_f(n,m)e^{-j2(\frac{2\pi m}{N})n} \tag{3.7}$$

From the above equation only the even samples can be recovered. Inserting $k-1 = n$ in the discrete WD of sampled function $f(t)$,

$$f(2n-1)f^*(1) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} W_f(n,m)e^{-j2(\frac{2\pi m}{N})(n-1)} \tag{3.8}$$

leads to the recovering of the odd samples.

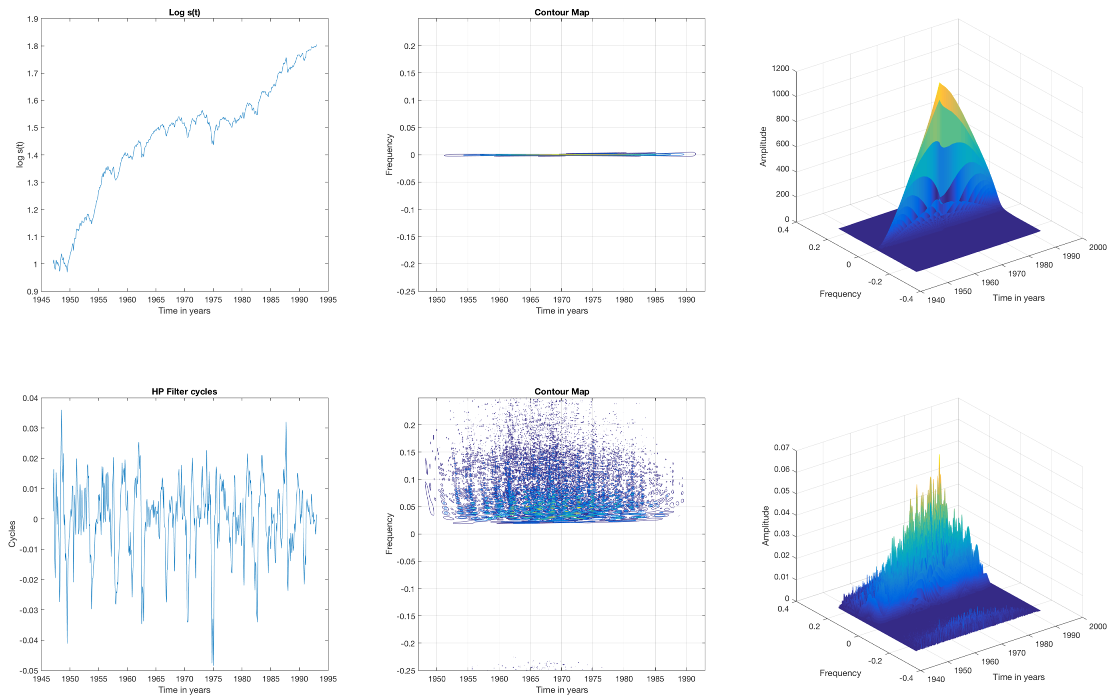


Figure 3.1: Wigner Distribution transform for log(s); where s denotes the sp500. Top row is the log(s), Wigner Distribution transform presentation in the contour and three dimensional mesh. Bottom row is the HP filter cycle of log(s) and its Wigner Distribution transform. The graph was created using Matlab code mywvd.m and it is attached in the Appendix B

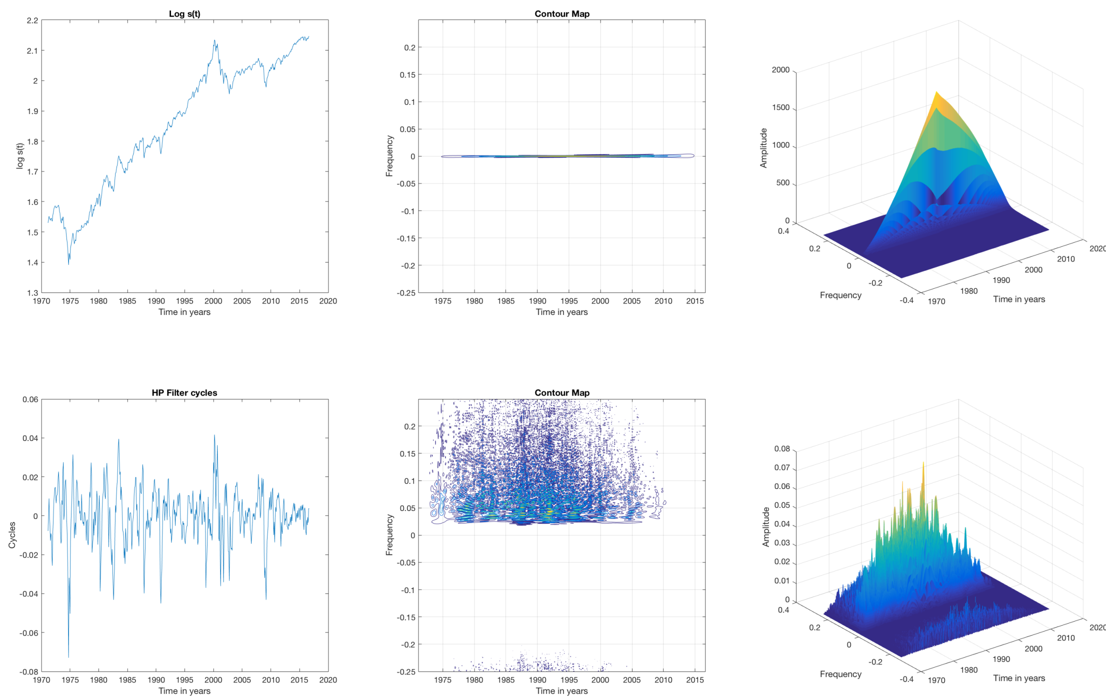


Figure 3.2: Wigner Distribution transform for log(s); where s denotes the NASDAQ index. Top row is the log(s), Wigner Distribution transform presentation in the contour and three dimensional mesh. Bottom row is the HP filter cycle of log(s) and its Wigner Distribution transform. The graph was created using Matlab code mywvd.m and it is attached in the Appendix B

CHAPTER IV

Time Frequency Distribution Series

4.1 Time Frequency Distribution Series

The main drawback of the Wigner Ville distribution is cross term interference and the cross term oscillates and localized.

Time Frequency Distribution Series (TFDS) introduced by Chen and Qian [**?**] as decomposition of the Wigner Ville distribution via the orthogonal like Gabor expansion. Let me walk through each step to attain the time frequency distribution series.

Let $g(t)$ be a normalized Gaussian function and is defined as below.

$$g(t) = \frac{1}{(\pi\sigma^2)^{0.25}} e^{-\frac{t^2}{2\sigma^2}} \tag{4.1}$$

The corresponding Wigner Ville Distribution (WVD) is given below.

$$WVD_g(t,\omega) = 2e^{-(\frac{t^2}{\sigma^2}+\sigma^2\omega^2)} \tag{4.2}$$

The $WVD_g(t,\omega)$ is centered at origin and it decays exponentially in both time and frequency domains. The contour plot of the $WVD_g(t,\omega)$ consists of concentric ellipses and it is given below.

WVD is time and frequency-shift invariant and let

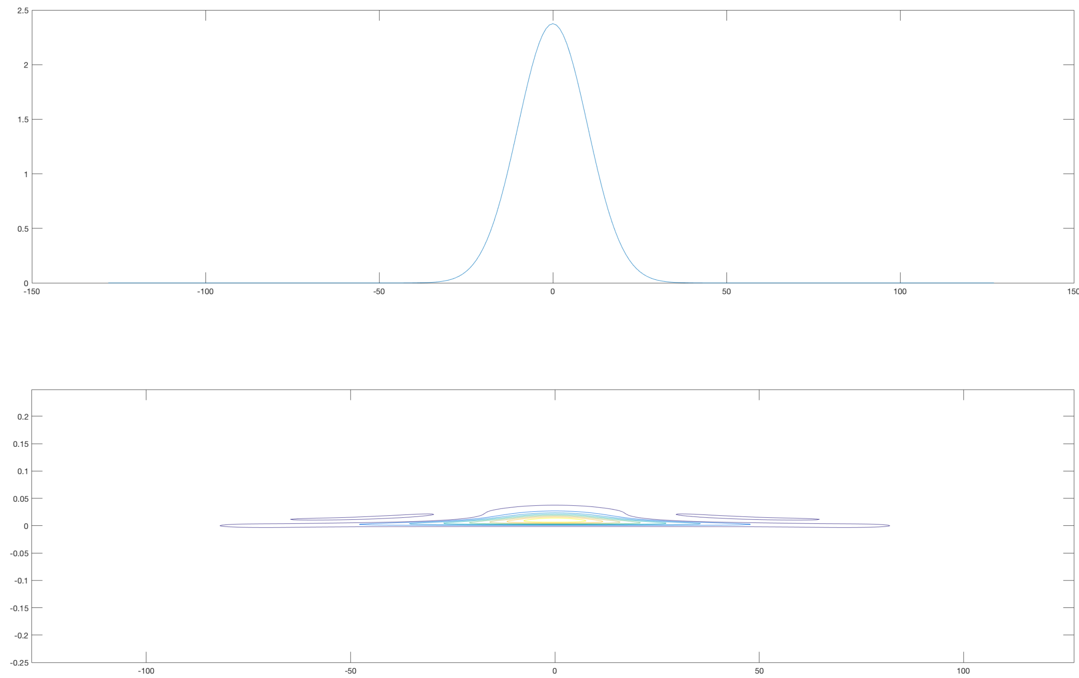


Figure 4.1: The Gaussian function with $\sigma = .1$ and its contour plot

$$h(t) = g(t - mT)e^{-jn\Omega t} \tag{4.3}$$

and the WVD of $h(t)$ is given below,

$$WVD_t(t, \omega) = 2e^{-[\frac{(t-mT)^2}{\sigma^2} + [\sigma(\omega - n\Omega)]^2]} \tag{4.4}$$

$$WVD_t(t, \omega) = WVD_g(t - mT, \omega - n\Omega) \tag{4.5}$$

Let $s(t) = h(t) + h'(t)$, then $WVD_s(t, \omega)$ is given as,

$$WVD_s(t, \omega) = WVD_t(t, \omega) + WVD_{h'}(t, \omega) + 2Re[WVD_{h,h'}(t, \omega)] \tag{4.6}$$

where $WVD_{h,h'}(t,\omega)$ is given by,

$$WVD_{h,h'}(t,\omega) = e^{j\omega_d t_\mu} H(t - t_\mu, \omega - \omega_\mu) \tag{4.7}$$

where

$$H(t,\omega) = 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[t_d\omega - \omega_d t]} \tag{4.8}$$

$$t_\mu = \frac{m+m'}{2}T, \omega_\mu = \frac{n+n'}{2}\Omega, t_d = (m-m')T, \omega_d = (n-n')\Omega \tag{4.9}$$

The $WVD_{h,h'}(t,\omega)$ has the same envelop as the $WVD_g(t,\omega)$ but is oscillating with ω_d in the time domain and t_d in the frequency domain. The location of $WVD_{h,h'}(t,\omega)$ is halfway between h and h'. The $2Re[WVD_{h,h1}(t,\omega)]$ is cross term. When a signal $s(t)$ can be decomposed as a linear combination of some elementary function $h(t)$ then the cross-terms can be controlled.

Recall from the previous chapter on Gabor Expansion, a given signal $s(t)$, the orthogonal-like Gabor expansion is defined as follows.

$$s(t) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{m,n} g(t - mT) e^{jn\Omega t} \tag{4.10}$$

The Gabor coefficients $C_{m,n}$ are determined by

$$C_{m,n} = \int s(t)\gamma^*_{m,n}(t)dt = \int s(t)\gamma^*(t - mT)e^{-jn\Omega t}dt = STFT(mT, n\Omega) \tag{4.11}$$

Taking Wigner-Ville distribution to $s(t)$ from the above equation yields,

$$WVD_s(t,\omega) = \sum_{m,n} \sum_{m',n'} C_{m,n} C_{m',n'} WVD_{h,h'}(t,\omega) \tag{4.12}$$

where

$$WVD_{h,h'}(t,\omega) = e^{j\omega_d t_\mu} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[t_d(\omega-\omega_d t]} \tag{4.13}$$

$$WVD_s(t,\omega) = \sum_{m,n}\sum_{m',n'} C_{m,n}C_{m',n'} e^{j\omega_d t_\mu} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[t_d\omega-\omega_d t]} \tag{4.14}$$

where t_d and ω_d reflect the degree of oscillation.

When $m = m'$ and $n = n'$, $t_\mu = T$, $\omega_\mu = \Omega$, $t_d = 0$ and $\omega_d = 0$ and substituting these values, $WVD_s(t,\omega)$ becomes,

$$WVD_s(t,\omega) = \sum_{m,n}\sum_{m',n'} C_{m,n}C_{m',n'} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} \tag{4.15}$$

$$WVD_s(t,\omega) = \sum_{m,n}\sum_{m',n'} C_{m,n}C_{m',n'} 2e^{-[\frac{(t-mT)^2}{\sigma^2}+[\sigma(\omega-n\Omega)^2)]} \tag{4.16}$$

$$C_{m,n}C_{m',n'}WVD_{h,h'}(t,\omega) = 2|C_{m,n}|^2 e^{-[\frac{(t-mT)^2}{\sigma^2}+[\sigma(\omega-n\Omega)^2)]} \tag{4.17}$$

When $m \neq m'$ or $n \neq n'$

$$C_{m,n}C_{m',n'}WVD_{h,h'}(t,\omega) = C_{m,n}C^*_{m',n'} e^{j\omega_d t_\mu} H(t-t_\mu, \omega-\omega_\mu) \tag{4.18}$$

Based on the decomposition of the Wigner-Ville distribution defined above, the Time Frequency Distribution Series (TFDS) is defined as follows.

$$TFDS_D(t,\omega) = \sum_{d=0}^{D} P_d(t,\omega) \tag{4.19}$$

Where $P_d(t,\omega)$ is the set of $WVD_{h,h'}(t,\omega)$ which have a similar contribution to the useful properties and similar influence to the cross terms in which $|m-m'|+|n-n'| = d$

$$P_d(t,\omega) = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C^*_{m',n'} WVD_{h,h'}(t,\omega) \tag{4.20}$$

Substituting the value of $WVD_{h,h'}(t,\omega)$ in the above equation, we get

$$P_d(t,\omega) = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C^*_{m',n'} e^{j\omega_d t_\mu} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[t_d\omega - \omega_d t]} \tag{4.21}$$

Substituting the value of t_d, ω_d, t_μ and ω_μ, we get

$$P_d(t,\omega) = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C^*_{m',n'} e^{j\frac{n+n'}{2}\Omega(m-m')T} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[(m-m')T\omega - (n-n')\Omega t]} \tag{4.22}$$

Both SP500 and NASDAQ index are discrete signal used for analysis. The P_d is further simplified for programming convenience.

The discrete Time Frequency Distribution Series is defined as follows.

$$DTFDS_D[i,k] = TFDS_d(t,\omega)|t = i\Delta t, \omega = \frac{2\pi k}{L\Delta t} \tag{4.23}$$

for $-\frac{L}{2} \leq k < \frac{L}{2}$

where $\frac{1}{\Delta t}$ denotes the sampling frequency. L denotes the length of the signal. The discrete time frequency distribution series can be summarized as

$$TFDS_d(i,k) = \sum_{d=0}^{D} P_d[i,k] \tag{4.24}$$

where

$$P_d[i,k] = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C_{m'n'} WVD_{h,h'}[i,k] \tag{4.25}$$

The $TFDS_D[i,k]$ is the sum of all $WVD_{h,h'}[i,k]$ in which the distance of the corresponding Gabor elementary functions $h_{m,n}[i]$ and $h_{m',n'}[i]$ is less than or equal to D. $WVD[i,k]$ is defined as a sampled Wigner-Ville distribution.

$$WVD_s[i,k] = WVD_s(t,\omega)|t = i\Delta t, \omega = \frac{2\pi k}{L\Delta t} \tag{4.26}$$

where Δt denotes the sampling interval. For the Gaussian functions, WVD is obtained by sampling the formula.

$$WVD_{h,h'}[i,k] = 2e^{-\sigma(i-\frac{m+m'}{2}\Delta M)^2-\frac{1}{\sigma}(k-\frac{n+n'}{2}\Delta N)^2} e^{j\frac{2\pi}{L}[(m-m')\Delta Mk+(n-n')\Delta Ni-\frac{n+n'}{2}\Delta N(m-m')\Delta M]} \tag{4.27}$$

We assume $\Delta t = 1$. $WVD_{h,h'}[i,k]$ in **??** is completely determined by the parameters of the Gabor expansion, such as $\Delta M, \Delta N, L$ and σ which are independent of the analyzed signal. Therefore, once $\Delta M, \Delta N, L$ and σ are determined, $WVD_{h,h'}[i,k]$ can be precomputed and saved in a table.

CHAPTER V

Color Chaos Model

5.1 Color Chaos

5.1.1 Chaos models

Chen claimed in [**?**], that correlation analysis and spectral analysis are complementary tools in the stationary time-series analysis. Random walk (white noise) has a zero correlation and a flat spectrum while sine wave has an infinite correlation and a sharp spectrum with zero width. Econometric models, such as ARCH and GARCH models with changing means and variance, are parametric models in the non-stationary stochastic approach. A generalized spectral approach is more useful in the studies of deterministic chaos. Based on the detail study of the paper [**?**],**Color Chaos** is defined as a time frequency representation as a nonparametric approach for generalized spectral analysis for evolutionary time series. In color chaos, the HP filter is applied for trend-cycle decomposition and time-variant filters in Gabor space for pattern recognition.

Discrete-time white chaos generalized by nonlinear difference equations is tractable analytically and from the needs of empirical analysis, the continuous-time color chaos generated by nonlinear differential equations is more capable of describing business cycles than white chaos, since fluctuations and recurrent patterns can be characterized by nonlinear oscillations with irregular amplitude and a narrow frequency band in the spectrum. Chen claimed in [**?**], the newly decoded deterministic signals from persistent business cycles reveal new sources of market uncertainty, such as changing growth-trend and shifting business cycles.

5.1.2 Role of time scale & reference trends in representation of business cycles

Problem: A distinctive problem in economic analysis is how to deal with growing trends in an aggregate economic time series. Both level and rate information are important when correlations are not short during business cycles.

Time Scale: Chaos theory in nonlinear dynamics emphasizes the role of history, because a nonlinear deterministic system is sensitive to its initial condition. The martingale theory of the stock market ignores the path-dependent information in the stock market. The challenges faced are:

1. Choosing an appropriate time sampling rate is often ignored in econometric analysis. Chaotic cycles in continuous time may look like noise if the sampling time internal is not small compared to its fundamental period of cycle. For example, annual economic data are not capable of revealing the frequency pattern of business cycle.

2. Numerically, a large time unit such as the annual time series can easily obscure a cyclic pattern in the correlation analysis of business cycles.

Reference Trend: How to choose a reference trend or a proper transformation to simplify the empirical pattern of business cycles? The core problem in economic analysis is not noise-smoothing but trend-defining in economic observation and decision making. There are two criteria in choosing the proper mathematical representation.

1. Mathematical reliability

2. Empirical verifiability.

Unlike experimental economics, macroeconomic time series are not reproducible in history. Traditional tests in econometric analysis have limited power in studies of an evolutionary economy containing deterministic components. A good fit of past data does not guarantee the ability for better future predictions.

There are two extreme approaches in econometric analysis: the trend-stationary (TS) approach of log-linear detrending (LLD) and the difference-stationary (DS) approach of first differencing (FD). A compromise between these two extremes is the Hodrick and Prescott (HP) filter.

In principle, a choice of observation reference is associated with a theory of economic dynamics. Log-linear detrending implies a constant exponential growth as shown in the figure 1.4.

The FD detrending produces a noisy picture that is predicted by the random-walk model with a constant drift (or the so-called unit-root model in econometric literature). Economically speaking, the FD detrending in econometrics implies that the level information in price indicators can be ignored in economic behavior. This assertion may conflict with many economic practices, since traders constantly watch economic trends, and no one will make an investment decision based only on the current rate of price changes. The error-correction model in econometrics tried to remedy the problem by addition some lagged-level information, such as using a one-year-before level as an approximation of the long-run equilibrium. Then comes the problem of what is the long run equilibrium in the empirical sense.

Option traders based on the Black-Scholes model find that it is extremely difficult to predict the mean, variance, and correlations from historical data.

A proper decomposition of trend and cycles may find an appropriate scheme to weigh the short-term and long-run impacts of economic movements in economic dynamics.

The essence of trend-cycle decomposition is finding an appropriate time window, or equivalently, a proper frequency window, for observing time-dependent movements. Log-linear detrending is a low-pass filter or wave detector, while first differencing is a high-pass filter or noise amplifier.

The main drawback of LLD detrending is its over-dependence on historical boundaries, while the DS series is to erratic from amplifying high-frequency noise.

HP filter has two advantages.

1. It is localized approach in detrending, with the problem of boundary dependence.

2. Frequency response is in the range of business cycles.

5.1.3 Instantaneous Autocorrelations and instantaneous frequency in Time-frequency representation

The concepts of instantaneous autocorrelation and instantaneous frequency are important in developing generalized spectral analysis. A symmetric window in a localized time interval is introduced in the instantaneous autocorrelation function of the bilinear Wigner distribution (WD); the corresponding time-dependent frequency (or simply time frequency) can be defined by the Fourier spectrum of its autocorrelations.

In the figure 5.3 the chaos of the SP500 is shown.

In the figure 5.1 the chaos of the SP500 is shown.

5.1.4 Time-Variant Filters in the Gabor Space

5.1.5 Characteristic Frequency andn Color Chaos

conclusion: We will see that introducing a time-frequency representation and the HP filter does reveal some historical features of business cycles that are not observable through the FD filter.

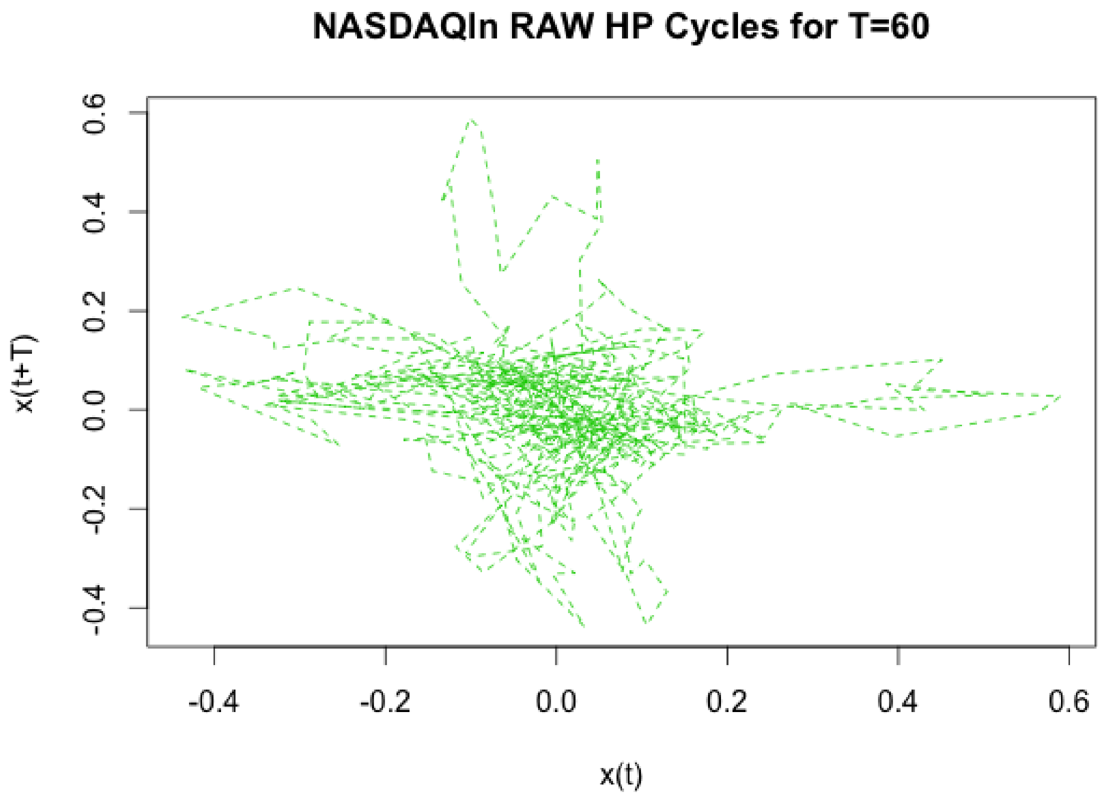


Figure 5.1: Phase Portrait for log(NASDAQ) unfiltered series $T = 60$. The graph was created by using Attractor.R (R program) and it is attached in the Appendix A

Data	η	$\nu(\%)$	CCgo	P_c	$\phi(\%)$	P_{dc}	λ^{-1}	μ
SP500	1.032	106.6	0.4141	NA	NA	NA	NA	
NASDAQ	1.0116	102.33	0.4735	NA	NA	NA	NA	

Table 5.1: Detrend Statistics on S&P 500

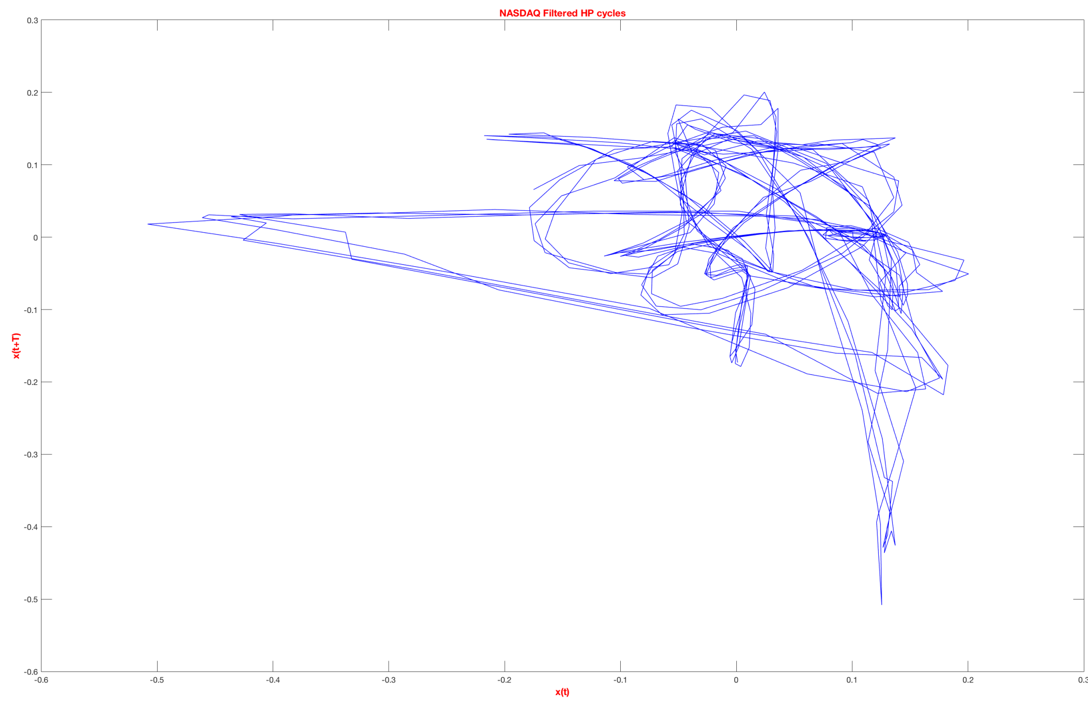


Figure 5.2: Phase Portrait for log(NASDAQ) unfiltered series $T = 60$. A pattern of strange attractor can be observed in the graph. The threshold value that depends on H ($H = 0.5$) has less significant impact to the pattern emergence whereas the size of m, n in $C(m, n)$, the Gabor Coefficient has significant impact to the pattern of strange attractor. The $16x16$ Gabor Coefficient was used. The graph was created by using myAttractor.m (Matlab program) and it is attached in the Appendix B.

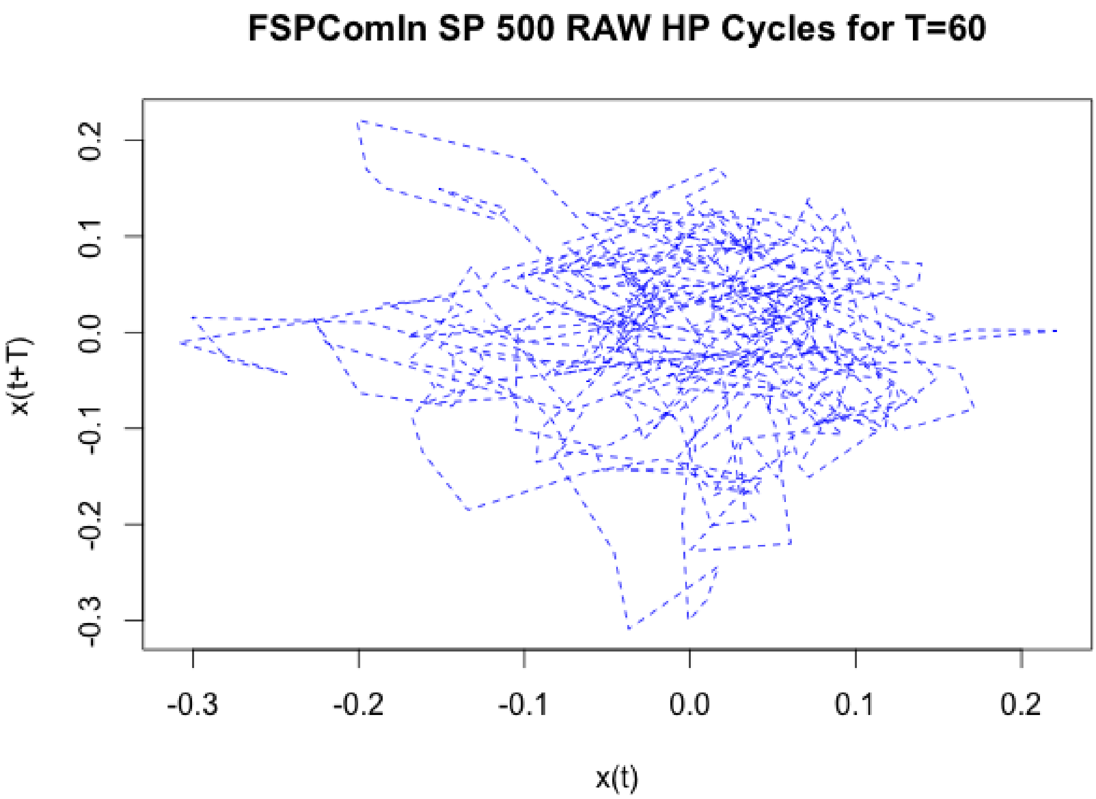


Figure 5.3: Phase Portrait for log(SP500) unfiltered series $T = 60$. The graph was created by using Attractor.R (R program) and it is attached in the Appendix A.

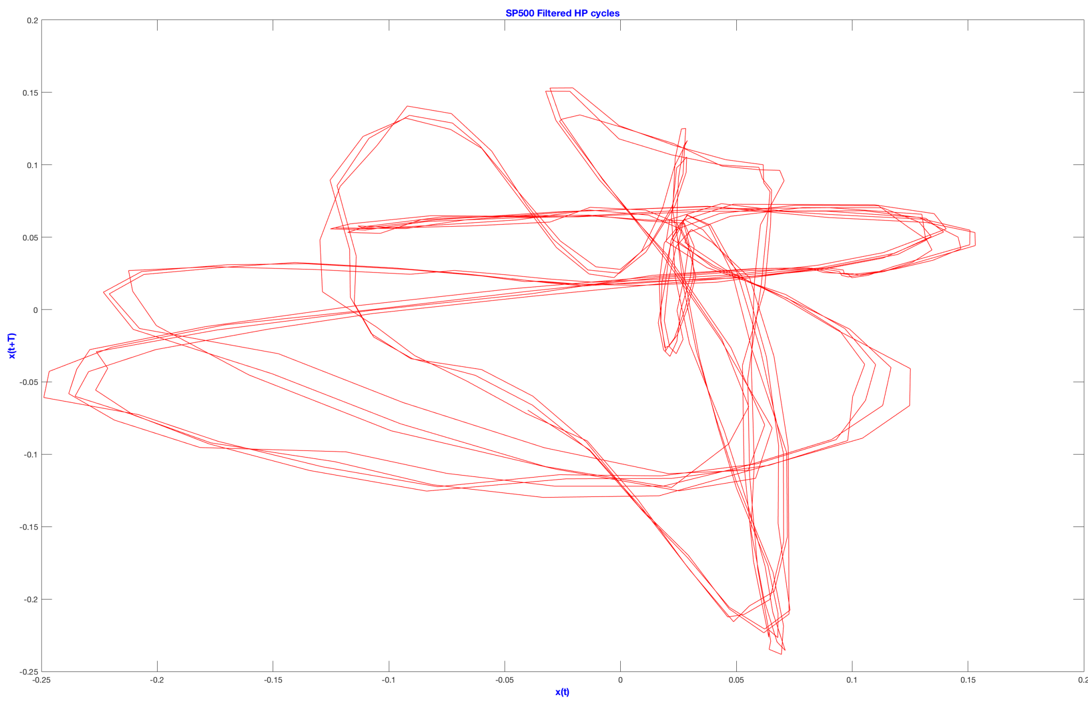
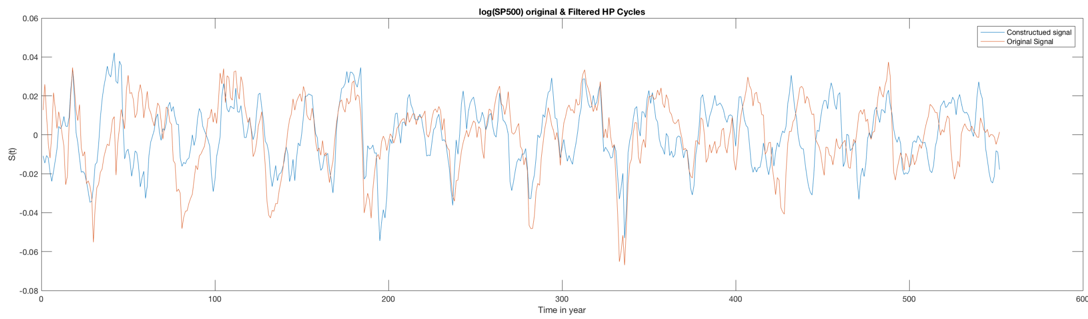


Figure 5.4: Phase Portrait for log(SP500) unfiltered series $T = 60$. A pattern of strange attractor can be observed in the graph. The threshold value that depends on H ($H = 0.5$) has less significant impact to the pattern emergence whereas the size of m, n in $C(m, n)$, the Gabor Coefficient has significant impact to the pattern of strange attractor. The $16x16$ Gabor Coefficient was used. The graph was created by using myAttractor.m (Matlab program) and it is attached in the Appendix B.



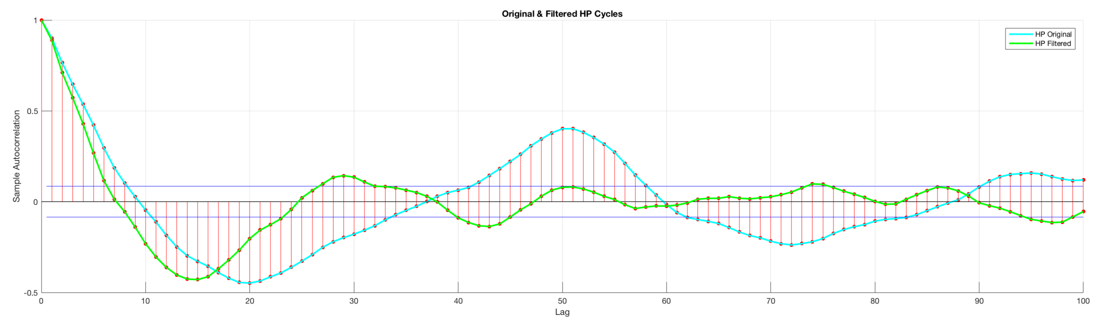


Figure 5.5: Fig a) The original and reconstructed time series of log(SP500) HP Cycles. Gabor Coefficients are created using the parameters - .$\Delta M = 8; \Delta N = 4; m = 16; n = 16; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. Fig b) Autocorelations of the original and reconstructed time series of log(SP500) HP cycle. The program used to create the graph is myreconstfromgabor.m and it is attached in the appendix B.

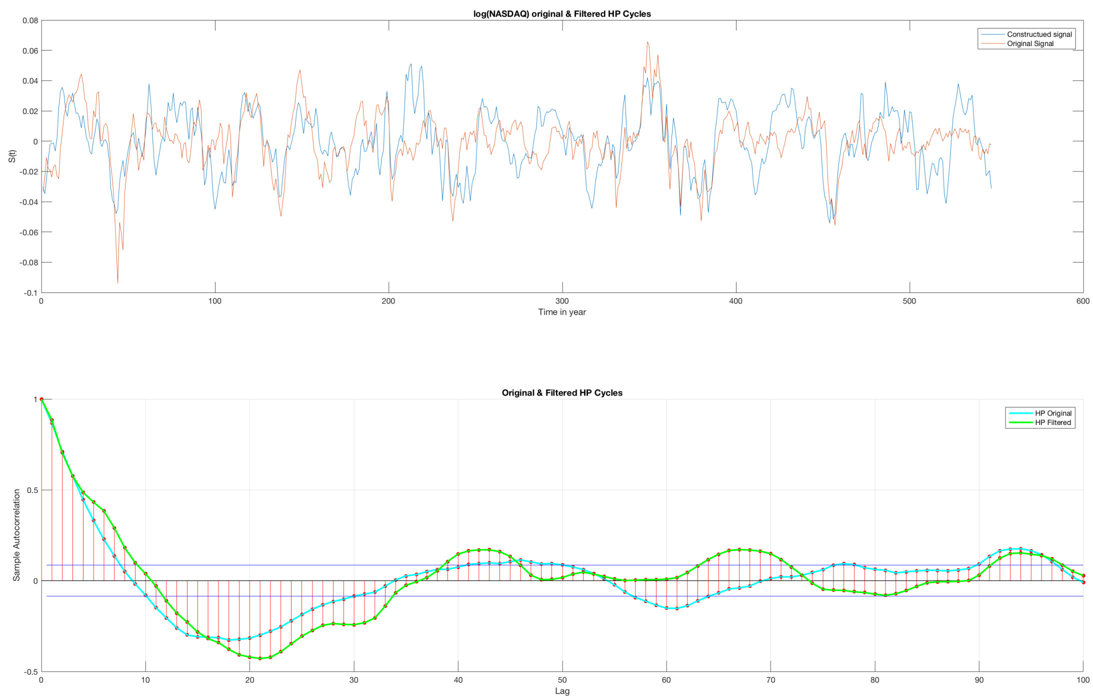


Figure 5.6: Fig a) The original and reconstructed time series of log(NASDAQ) HP Cycles. Gabor Coefficients are created using the parameters - .$\Delta M = 8; \Delta N = 4; m = 16; n = 16; \sigma = \sqrt{\frac{\Delta ML}{\Delta N 2\pi}}$. Fig b) Autocorelations of the original and reconstructed time series of log(NASDAQ) HP cycle. The program used to create the graph is myreconstfromgabor.m and it is attached in the appendix B.

APPENDICES

APPENDIX A

R code that are used to do analysis

```
fdplot <-function()
{
  #Program used to create the Difference stationary for NASDAQ & SP500 indexes
  # Praba Siva
  # praba@umich.edu
  # @prabasiva
  layout(matrix(c(1,1,2,2), 2, 2, byrow = TRUE))
  setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
  fspcom=read.table('fspcom.dat')
  dat = log(fspcom[,5])
  year=fspcom[,2]+1/12*fspcom[,3]
  t1=dat[1:length(dat)-1]
  t2=dat[2:length(dat)]
  plot(year[1:length(year)-1],t2-t1,type='l',
       main="Difference stationary of first Differencing of log(x(t))\nx(t) = S&P 500",
       xlab="Year",ylab="FD",col='blue',
       ylim=c(-.2,.2),cex.axis=1.1,cex.lab=1.5,lwd=2.2)
  setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
  dat <- read.csv(file="nasdaq_ready.csv",head=TRUE,sep=",")
  year=dat[,1]+1/12*dat[,2]
  dat=log(dat[,3])
  t1=dat[1:length(dat)-1]
  t2=dat[2:length(dat)]
  plot(year[1:length(year)-1],t2-t1,type='l',
       main="Difference stationary of first Differencing of log(x(t))\nx(t) = NASDAQ",
       xlab="Year",ylab="FD",col='red',
       ylim=c(-.2,.2),cex.axis=1.1,cex.lab=1.5,lwd=2.2)

}
```

```
llt <-function()
{
  #Program used to Log linear trend and cycles for SP500 & NASDAQ index
  # Praba Siva
  # praba@umich.edu
  # @prabasiva
setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
fspcom=read.table('fspcom.dat')
year=fspcom[,2]
tsfspcom=ts(log(fspcom[,5]),start=year[1],
            end=c(year[length(year)],12),frequency=12)
loglinear=stl(log(tsfspcom),s.window=5)
plot(loglinear,main="A Seasonal-Trend Decomposition of S&P 500",
     col='red')
setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
dat <- read.csv(file="nasdaq_ready.csv",head=TRUE,sep=",")
year=dat[,1]
dat=dat[,3]
tsnasdaq=ts(dat,start=year[1],
            end=c(year[length(year)]-1,12),frequency=12)
nloglinear=stl(log(tsnasdaq),s.window=5)
plot(nloglinear,main="A Seasonal-Trend Decomposition of NASDAQ",
     col="blue")
strend=loglinear$time.series[,2]
ntrend=nloglinear$time.series[,2]
plot(year[1:length(strend)],strend[1:length(strend)],
     col='blue',type='l',ylim=range(strend,ntrend))
lines(year[1:length(ntrend)],ntrend[1:length(ntrend)],
      col='red',type='l')
}
```

```

hpfilt <- function()
{
  #Program used to create the HP filter for lambda 80 & 800 for S&P 500 index
  #Load the file and invoke hpfilt()
  #Filename: hpfilter_slave.R
  # Praba Siva;praba@umich.edu; @prabasiva
library(mFilter)
library(latex2exp)
opar <- par(no.readonly=TRUE)
setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
fspcom=read.table('fspcom.dat')
dat = fspcom[,5]
syear=fspcom[,2]+1/12*fspcom[,3]
ldat=log(dat)
dat=ldat
dat.hp1 <- hpfilter(dat, freq=80,type="frequency",drift=FALSE)
dat.hp2 <- hpfilter(dat, freq=800,type="frequency",drift=FALSE)
par(mfrow=c(3,1),mar=c(3,3,2,1),cex=.8)
plot(syear,dat,  ylim=range(dat),
     main="S&P 500 Index ",
```

```
      col=2, ylab="",type='l',cex.axis=1.1,cex.lab=1.3,lwd=2.2)
plot(syear,dat.hp1$trend,  ylim=range(dat.hp1$trend),
      main="HP filter of S&P 500 Index: Trend,Lambda=80 ",
      col=4, xlab='praba siva', ylab="log(s(t))",type='l',cex.axis=1.1,cex.lab=1.3,lwd=2.2)
plot(syear,dat.hp1$cycle,  ylim=range(dat.hp1$cycle),
      main="HP filter of S&P 500 Index: Cycle,Lambda=80 ",
      col=3, ylab="",type='l',cex.axis=1.1,cex.lab=1.3,lwd=2.2)
par(mfrow=c(3,1),mar=c(3,3,2,1),cex=.8)
plot(syear,dat,  ylim=range(dat),
      main="S&P 500 Index ",
      col=2, ylab="",type='l',cex.axis=1.1,cex.lab=1.3,lwd=2.2)
plot(syear,dat.hp2$trend,  ylim=range(dat.hp1$trend),
      main="HP filter of S&P 500 Index: Trend,Lambda=800 ",
      col=4, xlab='praba siva', ylab="log(s(t))",type='l',cex.axis=1.1,cex.lab=1.3,lwd=2.2)
plot(syear,dat.hp1$cycle,  ylim=range(dat.hp2$cycle),
      main="HP filter of S&P 500 Index: Cycle,Lambda=800 ",
      col=3, ylab="",type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
par(opar)
opar <- par(no.readonly=TRUE)
setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
nasda <- read.csv(file="nasdaq_ready.csv",head=TRUE,sep=",")
nyear=nasda[,1]+1/12*nasda[,2]
nasda=nasda[,3]
lnasda=log(nasda)
nasda=lnasda
nasda.hp1 <- hpfilter(nasda, freq=80,type="frequency",drift=FALSE)
nasda.hp2 <- hpfilter(nasda, freq=800,type="frequency",drift=FALSE)
par(mfrow=c(3,1),mar=c(3,3,2,1),cex=.8)
plot(nyear,nasda,  xlab="Year",ylab="log s(t)",ylim=range(nasda),
      main="NASDAQ Index ",
      col=2, type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
plot(nyear,nasda.hp1$trend,  xlab='Year', ylim=range(nasda.hp1$trend),
      main="HP filter of NASDAQ Index: Trend,Lambda=80 ",
      col=4, type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
plot(nyear,nasda.hp1$cycle,  ylim=range(nasda.hp1$cycle), xlab="Year",
      main="HP filter of NASDAQ Index: Cycle,Lambda=80 ",
      col=3, type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
par(mfrow=c(3,1),mar=c(3,3,2,1),cex=.8)
plot(nyear,nasda,  ylim=range(nasda),
      main="NASDAQ Index ",
      col=2, ylab="log s(t)",type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
plot(nyear,nasda.hp2$trend,  ylim=range(nasda.hp1$trend),
      main="HP filter of NASDAQ Index: Trend,Lambda=800 ",
      col=4, xlab='Year', ylab="log(s(t))",type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
plot(nyear,nasda.hp1$cycle,  ylim=range(nasda.hp2$cycle),
      main="HP filter of NASDAQ Index: Cycle,Lambda=800 ",
      col=3, ylab="",type='l',cex.axis=1.1,cex.lab=1.5,lwd=2.2)
par(opar)
nasda.hp3 <- hpfilter(nasda, freq=1600,type="frequency",drift=FALSE)
nasda.hp4 <- hpfilter(nasda, freq=14400,type="frequency",drift=FALSE)
lambda=c(80,800,1600,14400);
c=1:4;
layout(matrix(c(1,1,2,2), 2, 2, byrow = TRUE))
plot(nyear,nasda.hp1$trend,ylab='Log NASDAQ',
      main=TeX('NASDAQ Trend HP filter with different $\\lambda$'),
      xlab='Year',col=c(1),type='l',cex.axis=1,cex.lab=1.3,lwd=2.2);
```

```
lines(nyear,nasda.hp2$trend,main=TeX('NASDAQ Trend HP filter with different $\\lambda$') ,
       col=c(2),type='l',cex.axis=1,cex.lab=1.1,lwd=2.2);
lines(nyear,nasda.hp3$trend,main=TeX('NASDAQ Trend HP filterwith different $\\lambda$'),
       col=c(3),type='l',cex.axis=1,cex.lab=1.1,lwd=2.2);
lines(nyear,nasda.hp4$trend,main=TeX('NASDAQ Trend HP filterwith different $\\lambda$'),
       col=c(4),type='l',cex.axis=1,cex.lab=1.1,lwd=2.2);
legend('topleft', legend=TeX(sprintf("$\\lambda = %d$", lambda)), lwd=1, col=c)
dat.hp3 <- hpfilter(dat, freq=1600,type="frequency",drift=FALSE)
dat.hp4 <- hpfilter(dat, freq=14400,type="frequency",drift=FALSE)
lambda=c(80,800,1600,14400);
c=1:4;
plot(syear,dat.hp1$trend,ylab='Log SP500',main=TeX('SP500 Trend HP filter with different ...
    $\\lambda$'),
      xlab='Year',col=c(1),type='l',cex.axis=1,cex.lab=1.3,lwd=2.2);
lines(syear,dat.hp2$trend,main=TeX('SP500 Trend HP filter with different $\\lambda$') ,
       col=c(2),type='l',cex.axis=1.1,cex.lab=1,lwd=2.2);
lines(syear,dat.hp3$trend,main=TeX('SP500 Trend HP filterwith different $\\lambda$'),
       col=c(3),type='l',cex.axis=1.1,cex.lab=1,lwd=2.2);
lines(syear,dat.hp4$trend,main=TeX('SP500 Trend HP filterwith different $\\lambda$'),
       col=c(4),type='l',cex.axis=1.1,cex.lab=1,lwd=2.2);
legend('topleft', legend=TeX(sprintf("$\\lambda = %d$", lambda)), lwd=1, col=c)

#Statistics of the HP Detrending
print('SP500 ')
print('HP Filter with Lambda = 80')
sprintf("Mean = %5f",mean(dat.hp1$cycle))
sprintf("SD = %5f",sd(dat.hp1$cycle))
sprintf("Variance = %5f",var(dat.hp1$cycle))
print('HP Filter with Lambda = 800')
sprintf("Mean = %5f",mean(dat.hp2$cycle))
sprintf("SD = %5f",sd(dat.hp2$cycle))
sprintf("Variance = %5f",var(dat.hp2$cycle))
print('HP Filter with Lambda = 1600')
sprintf("Mean = %5f",mean(dat.hp3$cycle))
sprintf("SD = %5f",sd(dat.hp3$cycle))
sprintf("Variance = %5f",var(dat.hp3$cycle))
print('HP Filter with Lambda = 14400')
sprintf("Mean = %5f",mean(dat.hp4$cycle))
sprintf("SD = %5f",sd(dat.hp4$cycle))
sprintf("Variance = %5f",var(dat.hp4$cycle))

print('NASDAQ ')
print('HP Filter with Lambda = 80')
xm=sprintf("Mean = %5f",mean(nasda.hp1$cycle))
xs=sprintf("SD = %5f",sd(nasda.hp1$cycle))
xv=sprintf("Variance = %5f",var(nasda.hp1$cycle))

print('HP Filter with Lambda = 800')
sprintf("Mean = %5f",mean(nasda.hp2$cycle))
sprintf("SD = %5f",sd(nasda.hp2$cycle))
sprintf("Variance = %5f",var(nasda.hp2$cycle))

print('HP Filter with Lambda = 1600')
sprintf("Mean = %5f",mean(nasda.hp3$cycle))
sprintf("SD = %5f",sd(nasda.hp3$cycle))
sprintf("Variance = %5f",var(nasda.hp3$cycle))
```

```
print('HP Filter with Lambda = 14400')
sprintf("Mean = %5f",mean(nasda.hp4$cycle))
sprintf("SD = %5f",sd(nasda.hp4$cycle))
sprintf("Variance = %5f",var(nasda.hp4$cycle))

}
```

```
loglinear<-function()
{
  #Program used to Log linear example
  #Filename:loglinear.R
  # Praba Siva
  # praba@umich.edu
  # @prabasiva
layout(matrix(c(1,1,2,2), 2, 2, byrow = TRUE))
t=0:50000;
plot(t,exp(t*-.0001),main=TeX('$\\beta_1 < 0$'),
      xlab='Time t', ylab=TeX('log Y =$\\beta_0+\\beta_1 t$'),
      type='l',cex.axis=1.1,cex.lab=.9,lwd=3,col='red')
plot(t,exp(t*.0001),main=TeX('$\\beta_1 > 0$'),
      xlab='Time t', ylab=TeX('log Y =$\\beta_0+\\beta_1 t$'),type='l',
      cex.axis=1.1,cex.lab=.9,lwd=3,col='blue')
par(opar)
}
```

```
ac<-function()
{
  #Program used to create AutoCorrelation Analysis for sample, SP500 & NASDAQ
  #Filename:AutoCorrelation.R
  # Praba Siva
  # praba@umich.edu
  # @prabasiva

library(mFilter);
library(latex2exp)
setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
fspcom=read.table('fspcom.dat')
dat = (fspcom[,5])
mort=log(dat)
year=fspcom[,2]+1/12*fspcom[,3]
le=length(dat)
x=mort[2:le]
y=mort[1:le-1]
diffxy=x-y
#plot(diffxy,type='l')
dur=1:length(year)
lmr=lm(mort~dur)
intercept=coef(lmr)[1]
slope=coef(lmr)[2]
```

```
dftrend=intercept+slope*dur
dfcycle=mort-dftrend
dfacf=acf(dfcycle,plot=FALSE,100);
hpf=hpfilter(mort,freq=14400)
layout(matrix(c(1,2,3,4), 4,1, byrow = TRUE))
color={'blue'}
ac1=acf(hpf$cycle, ci.type = "ma",plot=FALSE,100)
plot(year,mort,main='Log SP500 index',
     xlab='Year',ylab=TeX('log (SP500(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='red');
bc1=acf(diffxy,ci.type="ma",plot=FALSE,100)
plot(ac1,main='Autocorrelation of log SP500 HP Cycles'
     ,xlab='Lag',ylab='AC(1)')
lines(ac1$lag,ac1$acf,main='Autocorrelation of log SP500 HP Cycles',
      xlab='Lag',ylab='AC(1)',type='l',
      col='blue',cex.axis=1.1,cex.lab=1.1,lwd=3)
plot(bc1,main='Autocorrelation of log SP500 FD ',
     xlab='Lag',ylab='AC(1)')
lines(bc1$lag,bc1$acf,main='Autocorrelation of log SP500 FD',
      xlab='Lag',ylab='AC(1)',type='l',col='blue',lwd=3)
plot(dfacf,main='Autocorrelation of log-linear SP500  ',
     xlab='Lag',ylab='AC(1)')
lines(dfacf$lag,dfacf$acf,main='Autocorrelation of log-linear SP500 ',
      xlab='Lag',ylab='AC(1)',type='l',
      col='blue',lwd=3)
layout(matrix(c(1,2), 2,1, byrow = TRUE))
plot(year,mort,main='Log SP500 index',
     xlab='Year',ylab=TeX('log (SP500(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='red');
lines(year,dftrend,main='Trend of Log SP500 index using Log-linear',
      xlab='Year',ylab=TeX('log-linear(SP500(t))'),
      type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='blue');
legend("bottomright",c("Trend"),lty=c(1),lwd=c(2.5),col=c("blue"))
plot(year,dfcycle,main='Cycle of Log SP500 index using Log-linear',
     xlab='Year',ylab=TeX('log-linear(SP500(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='green');
layout(matrix(c(1,2), 2,1, byrow = TRUE))
plot(year,mort,main='Log SP500 index',
     xlab='Year',ylab=TeX('log (SP500(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='red');
plot(year[1:length(diffxy)],diffxy,
     main='Cycle of Log SP500 index using Log-linear trend',
     xlab='Year',ylab=TeX('log-linear(SP500(t))'),type='l',
     cex.axis=1.1,cex.lab=1.1,lwd=3,col='green');
sta.sp500=list(mean(dfacf$acf),sd(dfacf$acf),var(dfacf$acf),corrlength(dfacf),
               mean(ac1$acf),sd(ac1$acf),var(ac1$acf),corrlength(ac1),
               mean(bc1$acf),sd(bc1$acf),var(bc1$acf),corrlength(bc1));
layout(matrix(c(1,2,3,4), 4,1, byrow = TRUE))
setwd("/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program")
dat <- read.csv(file="nasdaq_ready.csv",head=TRUE,sep=",")
year=dat[,1]+1/12*dat[,2]
dat=dat[,3]
mort=log(dat)
le=length(dat)
x=mort[2:le]
y=mort[1:le-1]
```

```
diffxy=x-y
dur=1:length(year)
lmr=lm(mort¬dur)
intercept=coef(lmr)[1]
slope=coef(lmr)[2]
dftrend=intercept+slope*dur
dfcycle=mort-dftrend
dfacf=acf(dfcycle,plot=FALSE,100);
hpf=hpfilter(mort,freq=14400)
ac1=acf(hpf$cycle, ci.type = "ma",plot=FALSE,100)
bc1=acf(diffxy,ci.type="ma",plot=FALSE,100)
layout(matrix(c(1,2), 2,1, byrow = TRUE))
plot(year,mort,main='Log NASDAQ index',
     xlab='Year',ylab=TeX('log (NASDAQ(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='red');
lines(year,dftrend,main='Trend of Log NASDAQ index using Log-linear',
      xlab='Year',ylab=TeX('log-linear(NASDAQ(t))'),
      type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='blue');
legend("bottomright",c("Trend"),lty=c(1),lwd=c(2.5),col=c("blue"))
plot(year,dfcycle,main='Cycle of Log NASDAQ index using Log-linear',
     xlab='Year',ylab=TeX('log-linear(NASDAQ(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='green');
layout(matrix(c(1,2), 2,1, byrow = TRUE))
plot(year,mort,main='Log NASDAQ index',
     xlab='Year',ylab=TeX('log (NASDAQ(t))'),
     type='l',cex.axis=1.1,cex.lab=1.1,lwd=3,col='red');
plot(year[1:length(diffxy)],diffxy,
     main='Cycle of Log NASDAQ index using Log-linear trend',
     xlab='Year',ylab=TeX('log-linear(NASDAQ(t))'),type='l',
     cex.axis=1.1,cex.lab=1.1,lwd=3,col='green');
layout(matrix(c(1,2,3,4), 4,1, byrow = TRUE))
plot(year,mort,main='Log NASDAQ index',
     xlab='Year',ylab=TeX('log (NASDAQ(t))'),type='l',
     col='red',cex.axis=1.1,cex.lab=1.1,lwd=3);
plot(ac1,main='Autocorrelation of log NASDAQ HP Cycles',
     xlab='Lag',ylab='AC(1)')
lines(ac1$lag,ac1$acf,main='Autocorrelation of log NASDAQ HP Cycles',
      xlab='Lag',ylab='AC(1)',type='l',
      col='blue',cex.axis=1.1,cex.lab=1.1,lwd=3)
plot(bc1,main='Autocorrelation of log NASDAQ FD ',
     xlab='Lag',ylab='AC(1)')
lines(bc1$lag,bc1$acf,main='Autocorrelation of log NASDAQ FD',
      xlab='Lag',ylab='AC(1)',type='l',
      col='blue',cex.axis=1.1,cex.lab=1.1,lwd=3)
plot(dfacf,main='Autocorrelation of log-linear NASDAQ  ',
     xlab='Lag',ylab='AC(1)')
lines(dfacf$lag,dfacf$acf,main='Autocorrelation of log-linear NASDAQ ',
      xlab='Lag',ylab='AC(1)',type='l',col='blue',lwd=3)
layout(matrix(c(1,2,3,4,5,6), 3, 2, byrow = TRUE))
#par(mfrow=c(2,1),mar=c(3,3,2,1),cex=.8)
x=seq(-15,15,.1);
y=sin(x)
ac1=acf(y,lag.max=100,plot=FALSE);
plot(x,y,main='Sin wave',xlab='T',ylab='Sin(t)',type='l',
     col='red',cex.axis=1.1,cex.lab=1.1,lwd=3)
plot(ac1,main='Autocorrelation of Sin wave',xlab='Lag',ylab='AC(1)',
```

```
      cex.axis=1.1,cex.lab=1.1,lwd=.2)
lines(ac1$lag,ac1$acf,type='l',col='blue',lwd=2)
x=seq(-15,15,.1);
y=x^2+x^3
ac1=acf(y,lag.max=100,plot=FALSE);
plot(x,y,main='Polynomial',
      xlab='T',ylab=TeX('y=x^3(t)+x^2(t)'),type='l',col='red',
      cex.axis=1.1,cex.lab=1.1,lwd=3)
plot(ac1,main='Autocorrelation of Polynomial',
      xlab='Lag',ylab='AC(1)',cex.axis=1.1,cex.lab=1.5,lwd=.2)
lines(ac1$lag,ac1$acf,type='l',col='blue',lwd=2)
x=seq(-15,15,.1);
y=sin(x)*rnorm(length(x),mean=0,sd=1)
ac1=acf(y,lag.max=100,plot=FALSE);
plot(x,y,main='Sin wave with random noise',
      xlab='t',
      ylab=TeX('Sin(t) * r($\\mu=0 ,\\sigma^2=1)'),type='l',col='red',
      cex.axis=1.1,cex.lab=1.1,lwd=2)
plot(ac1,main='Autocorrelation of Sin wave with random noise',
      xlab='Lag',ylab='AC(1)',cex.axis=1.1,cex.lab=1.5,lwd=3)
lines(ac1$lag,ac1$acf,type='l',col='blue',lwd=.2)
corrlength(ac1)
sta.nasdaq=list(mean(dfacf$acf),sd(dfacf$acf),var(dfacf$acf),corrlength(dfacf),
                mean(ac1$acf),sd(ac1$acf),var(ac1$acf),corrlength(ac1),
                mean(bc1$acf),sd(bc1$acf),var(bc1$acf),corrlength(bc1))
print("Dtrend statistics for SP500")
print(matrix(sta.sp500,nrow=4))
print("Dtrend statistics for NASDAQ")
print(matrix(sta.nasdaq,nrow=4))
}

corrlength <- function(acfvector)

{

  ind=min(which(acfvector$acf<0));

  return ...
     ((abs(acfvector$acf[ind])+abs(acfvector$acf[ind-1])/10)*(abs(acfvector$acf[ind-1]))+ind-1)

}
```

APPENDIX B

Matlab code that are used to do analysis

```
function DrawSinFourierGraph()
Fs = 1000;                    % Sampling frequency
T = 1/Fs;                     % Sampling period
L = 200;                        % Length of signal
t = (0:L-1)*T;                % Time vect
f = [50,150,300];
x1 = sin(2*pi*f(1)*t);          % First row wave
x2 = sin(2*pi*f(2)*t);         % Second row wave
x3 = sin(2*pi*f(3)*t);         % Third row wave

X = [x1; x2; x3;];

figure;

for i = 1:3
    g=subplot(3,2,i+i-1)
    plot(t(1:L),X(i,1:L),'r','LineWidth',1)
    ylabel('sin(2\pift)','FontSize',16,'FontWeight','bold')
    xlabel('\fontname{Helvetica} Time','FontSize',16,'FontWeight','bold')
    title(['A sin wave of frequency f= ',num2str(f(i)),' in the Time ...
        Domain'],'FontSize',18,'FontWeight','bold')
    p=get(g,'position');
    p(1)=.7*p(1);
    p(4)=1.1*p(4);
    set(g,'position',p);
end


n = 2^nextpow2(L);
dim = 2;
Y = fft(X,n,dim);
P2 = abs(Y/n);
```

```
P1 = P2(:,1:n/2+1);
P1(:,2:end-1) = 2*P1(:,2:end-1);


for i=1:3
    g=subplot(3,2,i*2);
    plot(0:(Fs/n):(Fs-Fs/n),P2(i,1:n),'b','LineWidth',1)
    title(['Sin wave of frequency  ',num2str(f(i)), ' in the Frequency ...
        Domain'],'FontSize',18,'FontWeight','bold')
    ylabel(' |F(s)|','FontSize',16,'FontWeight','bold')
    xlabel('Freqency','FontSize',16,'FontWeight','bold')
    p=get(g,'position');
    p(1)=.9*p(1);
    p(4)=1.1*p(4);
    set(g,'position',p);
end
```

```
function SPNASDAQFourier()
Fs = 1000;            % Sampling frequency
T = 1/Fs;             % Sampling period
L = 1000;             % Length of signal
t = (0:L-1)*T;        % Time vector
S = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
X = S + 2*randn(size(t));
g=subplot(5,2,1);
plot(1000*t(1:50),X(1:50),'r','LineWidth',1);
title('Signal Corrupted with Zero-Mean Random Noise','FontSize',18,'FontWeight','bold')
xlabel('t (milliseconds)','FontSize',16,'FontWeight','bold')
ylabel('f(t)','FontSize',16,'FontWeight','bold')
Y = fft(X);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
g=subplot(5,2,2);
plot(f,P1,'b','LineWidth',1)
title('Single-Sided Amplitude Spectrum of f(t)','FontSize',16,'FontWeight','bold')
xlabel('w','FontSize',16,'FontWeight','bold')
ylabel('|F(w)|','FontSize',16,'FontWeight','bold')
g=subplot(5,2,3);
plot(1000*t(1:50),S(1:50),'r','LineWidth',1);
title('Signal with Zero-Mean Random Noise','FontSize',18,'FontWeight','bold')
xlabel('t (milliseconds)','FontSize',16,'FontWeight','bold')
ylabel('f(t)','FontSize',16,'FontWeight','bold')
Y = fft(S);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
g=subplot(5,2,4);
plot(f,P1,'b','LineWidth',1)
title('Single-Sided Amplitude Spectrum of f(t)','FontSize',18,'FontWeight','bold')
xlabel('w','FontSize',16,'FontWeight','bold')
```

```
ylabel('|F(w)|','FontSize',16,'FontWeight','bold')
Fs = 100;            % Sampling frequency
t = -0.5:1/Fs:0.5;  % Time vector
L = length(t);       % Signal length
X = 1/(4*sqrt(2*pi*0.01))*(exp(-t.^2/(2*0.01)));
g=subplot(5,2,5)
plot(t,X,'r','LineWidth',1)
title('Gaussian Pulse in Time Domain','FontSize',18,'FontWeight','bold')
xlabel('Time (t)','FontSize',16,'FontWeight','bold')
ylabel('f(t)','FontSize',16,'FontWeight','bold')
n = 2^nextpow2(L);
Y = fft(X,n);
f = Fs*(-((n/2)/n):(n/2))/n;
P = abs(Y/n);
g=subplot(5,2,6)
plot(f,P(1:n/2+1),'b','LineWidth',1)
title('Gaussian Pulse in Frequency Domain','FontSize',18,'FontWeight','bold')
xlabel('Frequency (w)','FontSize',16,'FontWeight','bold')
ylabel('|F(s)|','FontSize',16,'FontWeight','bold')
dat=csvread('/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program/nasdaq_mat.csv');
year=dat(:,1);
month=dat(:,2);
naq=dat(:,3);
naq=log(naq);
year=year+month/12;
g=subplot(5,2,7)
plot(year,naq,'r','LineWidth',1);
title('Log Nasdaq','FontSize',18,'FontWeight','bold');
xlabel('Year.month','FontSize',16,'FontWeight','bold');
ylabel('Log Nasdaq','FontSize',16,'FontWeight','bold');
Fs = 1000;
[L,tp]=size(naq)
n = 2^nextpow2(L);
Y = fft(naq,n);
f = Fs*(-((n/2)/n):(n/2))/n;
P = abs(Y/n);
g=subplot(5,2,8)
%plot(f,P(1:n/2+1),'b','LineWidth',1)
plot(1:50,P(1:50),'b','LineWidth',1)
title('Nasdaq in Frequency Domain','FontSize',18,'FontWeight','bold')
xlabel('Frequency (w)','FontSize',16,'FontWeight','bold')
ylabel('|F(w)|','FontSize',16,'FontWeight','bold')
dat=readtable('/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program/fspcom.dat');
[maxx,maxy]=size(dat);
sp500=table2array(dat(1:maxx,5));
year=(table2array(dat(1:maxx,2)));
month=(table2array(dat(1:maxx,3)));
year=year+month/12
sp500=log(sp500)
g=subplot(5,2,9)
plot(year,sp500,'r','LineWidth',1);
title('Log S&P 500','FontSize',18,'FontWeight','bold');
xlabel('Year.month','FontSize',16,'FontWeight','bold');
ylabel('Log S&P 500','FontSize',16,'FontWeight','bold');
[L,tp]=size(sp500)
n = 2^nextpow2(L);
```

```
Y = fft(sp500,n);
f = Fs*(-((n/2)/n):(n/2))/n;
P = abs(Y/n);
g=subplot(5,2,10)
%plot(f,P(1:n/2+1),'b','LineWidth',1);
plot(1:50,P(1:50),'b','LineWidth',1);
title('S&P in Frequency Domain','FontSize',16,'FontWeight','bold')
xlabel('Frequency (w)','FontSize',16,'FontWeight','bold')
ylabel('|F(w)|','FontSize',16,'FontWeight','bold')
```

```
function ghamwin()
s={'Gaussian Window' 'Hamming Window'};
wlen=25;
figure;
for fla=0:1
    if fla<1
    % form a periodic hamming window
        win = hamming(wlen, 'periodic');
    else
        win=gausswin(wlen)
    end
    g=subplot(1,2,1+fla);
    plot(abs(win),'r','LineWidth',1);
    xlabel('n (N=25)','FontSize',16,'FontWeight','bold')
    if fla ==0
        title('\sigma = 2.5 Gaussian Window ...
            function','interpreter','Tex','Fontsize',16,'FontWeight','bold')
      ylabel('e^{-n^2/2\sigma^2}','interpreter','Tex','FontSize',20)
    else
        title([s(fla+1),'function'],'FontSize',16,'FontWeight','bold')
        ylabel('0.54-0.46cos(2\pi*n/N)','interpreter','Tex','FontSize',20)
    end
end
```

```
function drawSTFTEg()
s={'Gaussian Window' 'Hamming Window'};
wlen=25;
hopsize=25;
retno =1;
Ff = 500;
for fla=0:1
    Fs = 1000;            % Sampling frequency
    T = 1/Fs;             % Sampling period
    L = 1000;             % Length of signal
    t = (0:L-1)*T;        % Time vector
    S = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
    X = S + 2*randn(size(t));
    figure;
    g=subplot(2,2,1);
    plot(1000*t(1:100),X(1:100),'r','LineWidth',1);
```

```
    L = length(X);
    title('Signal Corrupted with Zero-Mean Random Noise','FontSize',18,'FontWeight','bold')
    xlabel('t (milliseconds)','FontSize',16,'FontWeight','bold')
    ylabel('f(t)','FontSize',16,'FontWeight','bold')
    [Y,ft,tt]=stft2(X,wlen,hopsize,retno,Ff,fla);
    %Y=fftshift(Y);
    P=abs(Y/L);
    g=subplot(2,2,2);
    plot(tt,P,'b','LineWidth',1);
    title(['STFT using', s(fla+1)],'FontSize',16,'FontWeight','bold')
    xlabel('\omega','FontSize',16,'FontWeight','bold')
    ylabel('|F(\omega,\tau)|','interpreter','Tex','FontSize',16,'FontWeight','bold')
    g=subplot(2,2,3);
    plot(1000*t(1:100),S(1:100),'r','LineWidth',1);
    L = length(X);
    title('Signal Corrupted with Zero-Mean Random Noise','FontSize',18,'FontWeight','bold')
    xlabel('t (milliseconds)','FontSize',16,'FontWeight','bold')
    ylabel('f(t)','FontSize',16,'FontWeight','bold')
    [Y,ft,tt]=stft2(S,wlen,hopsize,retno,Ff,fla);
    %Y=fftshift(Y);
    P=abs(Y/L);
    g=subplot(2,2,4);
    plot(tt,P,'b','LineWidth',1);
    title(['STFT using',s(fla+1)],'FontSize',16,'FontWeight','bold')
    xlabel('\omega','interpreter','Tex','FontSize',16,'FontWeight','bold')
    ylabel('|F(\omega,\tau)|','interpreter','Tex','FontSize',16,'FontWeight','bold')

end
```

```
function [stft, f, t] = stft2(x, wlen, h, nfft, fs,flag)
% function: [stft, f, t] = stft(x, wlen, h, nfft, fs)
% x - signal in the time domain
% wlen - length of the hamming window
% h - hop size
% nfft - number of FFT points
% flag = 1 for Gaussian window or 0 for Hamming window
% fs - sampling frequency, Hz
% f - frequency vector, Hz
% t - time vector, s
% stft - STFT matrix (only unique points, time across columns, freq across rows)
% represent x as column-vector if it is not
if size(x, 2) > 1
    x = x';
end
% length of the signal
xlen = length(x);
if flag<1
% form a periodic hamming window
win = hamming(wlen, 'periodic');
else
win=gausswin(wlen)
end
% form the stft matrix
```

```
rown = ceil((1+nfft)/2);            % calculate the total number of rows
coln = 1+fix((xlen-wlen)/h);        % calculate the total number of columns
stft = zeros(rown, coln);           % form the stft matrix

% initialize the indexes
indx = 0;
col = 1;

% perform STFT
while indx + wlen ≤ xlen
    % windowing
    xw = x(indx+1:indx+wlen).*win;

    % FFT
    X = fft(xw, nfft);

    % update the stft matrix
    stft(:, col) = X(1:rown);

    % update the indexes
    indx = indx + h;
    col = col + 1;
end

% calculate the time and frequency vectors
t = (wlen/2:h:wlen/2+(coln-1)*h)/fs;
f = (0:rown-1)*fs/nfft;

end
```

```
function [index,year]= getData(flag)
if flag == 1
    dat=readtable('/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program/fspcom.dat');
    [maxx,maxy]=size(dat);
    index=table2array(dat(1:maxx,5));
    year=(table2array(dat(1:maxx,2)));
    month=(table2array(dat(1:maxx,3)));
    index=log(index);
    year=year+month/12;
else
    dat=csvread('/Users/sivasp1/Documents/2016/Personal/Praba/MATH599/program/nasdaq_mat.csv');
    year=dat(:,1);
    month=dat(:,2);
    index=dat(:,3);
    index=log(index);
    year=year+month/12;
end
```

```
function mywvd()

```

```
%Program used to Wigner Distribution for NASDAQ & SP500 indexes
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: mywvd.m

close all;
clear all;
[sp500,syear]=getData(1);
sp500=log(sp500);
[naq,nyear]=getData(2);
naq=log(naq);

for step = 1:2

    if step == 2
        sp500=naq;
        syear=nyear;
    end;

    figure;

    [s2,s1]=hpfilter(sp500,1600);
    subplot(2,3,1);
    plot(syear,sp500);
    xlabel('Time in years');
    ylabel('log s(t)');
    title('Log s(t)');

    [wd,freq]=wig2(sp500);
    subplot(2,3,2);
    contour(syear,freq, abs(wd'),8), grid on
    xlabel('Time in years');
    ylabel('Frequency');
    title('Contour Map');

    subplot(2,3,3);
    mesh(syear,freq,abs(wd'));
    xlabel('Time in years');
    ylabel('Frequency');
    zlabel('Amplitude');

    subplot(2,3,4);
    plot(syear,s1);
    xlabel('Time in years');
    ylabel('Cycles');
    title('HP Filter cycles');

    [wd,freq]=wig2(s1);
    subplot(2,3,5);
    contour(syear,freq, abs(wd'),8), grid on
    xlabel('Time in years');
    ylabel('Frequency');
    title('Contour Map');

    subplot(2,3,6);
```

```
    mesh(syear,freq,abs(wd'));
    xlabel('Time in years');
    ylabel('Frequency');
    zlabel('Amplitude');

end;
```

```
function mygabor()
%Program used to Gabor Coefficients for SP500 & NASDAQ
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: mygabor.m
close all;
clear all;
[sp500,syear]=getData(1);
[naq,nyear]=getData(2);
    Δm = 8;
    %M=16;
    M=50
    Δn=4;
    %nn=32;
    nn=100;


    [s2,s1]=hpfilter(sp500,1600);


    s1=s1';
    L=length(s1);
    t=1:L;
    N=L/2;
    nn2=nn/2;
    sigma=sqrt((Δm*L)/(Δn * 2 * pi));
    c=nthroot(pi*sigma*sigma,-4);
    h0 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
    h = @(ii) h0( mod(ii + N, L)-N);
    for m = 1:M
        for n = 1:nn2
            c1(m, n)= sum(s1.*h(mod(t - m*Δm,L)).*exp(-2*pi*i*Δn*n*t/L));
        end

    end

    [s2,s1]=hpfilter(naq,1600);
    s1=s1';
    L=length(s1);
    t=1:L;
    N=L/2;
    nn2=nn/2;
    sigma=sqrt((Δm*L)/(Δn * 2 * pi));
    c=nthroot(pi*sigma*sigma,-4);
    h1 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
```

```
    h2 = @(ii) h1( mod(ii + N, L)-N);
    for m = 1:M
        for n = 1:nn2
            c2(m, n)= sum(s1.*h2(mod(t - m*∆m,L)).*exp(-2*pi*i*∆n*n*t/L));
        end
    end
    subplot(1,2,1)
    surf(abs(c1));
    %Change the text location based on the m & n.
    %For m=n=16, text(-5,0.4...
    %For m=n=50  text
    if M==16
        text(-5, 0.4, 'Fig a: Gabor Coefficient for ...
            log(sp500)','FontSize',16,'FontWeight','bold','Color','r')
    else
        text(-15, 0.4, 'Fig a: Gabor Coefficient for ...
            log(sp500)','FontSize',16,'FontWeight','bold','Color','r')
    end;
    xlabel('Time','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'xlabel');
    set(h,'rotation',30)
    ylabel('Frequency','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'ylabel');
    set(h,'Position',get(h,'Position') +[2 4 0])
    set(h,'rotation',140)
    zlabel('|C(m,n|^2','FontSize',12,'FontWeight','bold','Color','b')

    subplot(1,2,2);
    surf(abs(c2));
    colormap hsv;
    if M==16
        text(-5, 0.4, 'Fig a: Gabor Coefficient for ...
            log(nasdaq)','FontSize',16,'FontWeight','bold','Color','r')
    else
        text(-15, 0.4, 'Fig a: Gabor Coefficient for ...
            log(nasdaq)','FontSize',16,'FontWeight','bold','Color','r')
    end;
    xlabel('Time','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'xlabel');
    set(h,'rotation',30)
    ylabel('Frequency','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'ylabel');
    set(h,'Position',get(h,'Position') +[2 4 0])
    set(h,'rotation',140)
    zlabel('|C(m,n|^2','FontSize',12,'FontWeight','bold','Color','b')
end
```

```
function myfiltgabor()
%Program used to Gabor Coefficients for SP500 & NASDAQ
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: mygabor.m
```

```
close all;
clear all;
[sp500,syear]=getData(1);
[naq,nyear]=getData(2);
    Δm = 8;
    M=16;
    %M=50;
    Δn=4;
    nn=32;
    %nn=100;
    thrcont=3;
    [s2,s1]=hpfilter(sp500,1600);
    s1=s1';
    L=length(s1);
    t=1:L;
    N=L/2;
    nn2=nn/2;
    sigma=sqrt((Δm*L)/(Δn * 2 * pi));
    c=nthroot(pi*sigma*sigma,-4);
    h0 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
    h = @(ii) h0( mod(ii + N, L)-N);
    for m = 1:M
        for n = 1:nn2
            c1(m, n)= sum(s1.*h(mod(t - m*Δm,L)).*exp(-2*pi*i*Δn*n*t/L));
        end

    end
    [s2,s1]=hpfilter(naq,1600);
    s1=s1';
    L=length(s1);
    t=1:L;
    N=L/2;
    nn2=nn/2;
    sigma=sqrt((Δm*L)/(Δn * 2 * pi));
    c=nthroot(pi*sigma*sigma,-4);
    h1 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
    h2 = @(ii) h1( mod(ii + N, L)-N);
    for m = 1:M
        for n = 1:nn2
            c2(m, n)= sum(s1.*h2(mod(t - m*Δm,L)).*exp(-2*pi*i*Δn*n*t/L));
        end
    end

[m,n]=size(c1);
thr=max(abs(c1));
spmask = (max(thr)-min(thr))/thrcont;
for k = 1:m
    for j = 1:n
        if abs(c1(k,j)) < spmask
            sp500maskmatrix(k,j)=0;
        else
            sp500maskmatrix(k,j)=1;
        end;
    end;
end;

```

```
%DRAW THRESHOLD PRESENTATION & MASK OPERATOR
figure;
subplot(2,1,1);
plot(abs(c1),'LineWidth',2);
xlabel('m');
ylabel('|C(m,n)|');
title('Time Section of Gabor Distribution for SP500');
hold on;
li(1:m)=spmask;
p1=plot(li,'LineWidth',6,'Color','b');
legend(p1,'Threshold');

[m,n]=size(c2);
thr(1:m)=max(abs(c2(1:m,:)));

nasmask = (max(thr)-min(thr))/thrcont;
for k = 1:m
    for j = 1:n
        if abs(c2(k,j)) < nasmask
            nasmaskmatrix(k,j)=0;
        else
            nasmaskmatrix(k,j)=1;
        end;
    end;
end;
subplot(2,1,2);
plot(abs(c2),'LineWidth',2);
xlabel('m');
ylabel('|C(m,n)|');
title('Time Section of Gabor Distribution for NASDAQ');
hold on;
li(1:m)=nasmask;
p1=plot(li,'LineWidth',6,'Color','b');
legend(p1,'Threshold');
c1=c1.*sp500maskmatrix;
c2=c2.*nasmaskmatrix;
figure;
subplot(2,1,1);
plot(abs(c1),'LineWidth',2);
xlabel('m');
ylabel('|C(m,n)|');
title('Time Section of Masked Gabor Distribution for SP500 ');
hold on;
li(1:m)=spmask;
p1=plot(li,'LineWidth',6,'Color','b');
legend(p1,'Threshold');

subplot(2,1,2);
plot(abs(c2),'LineWidth',2);
xlabel('m');
ylabel('|C(m,n)|');
title('Time Section of Masked Gabor Distribution for NASDAQ');
hold on;
li(1:m)=nasmask;
p1=plot(li,'LineWidth',6,'Color','b');
legend(p1,'Threshold');
```

```
figure;

    subplot(1,2,1)
    surf(abs(c1));
    colormap hsv;
    %Change the text location based on the m & n.
    %For m=n=16, text(-5,0.4...
    %For m=n=50  text
    if M==16
        text(-5, 0.4, 'Fig a: Filtered Gabor Coefficient for ...
            log(sp500)','FontSize',16,'FontWeight','bold','Color','r')
    else
        text(-15, 0.4, 'Fig a: Filtered Gabor Coefficient for ...
            log(sp500)','FontSize',16,'FontWeight','bold','Color','r')
    end;
    xlabel('Time','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'xlabel');
    set(h,'rotation',30)
    ylabel('Frequency','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'ylabel');
    set(h,'Position',get(h,'Position') +[2 4 0])
    set(h,'rotation',140)
    zlabel('|C(m,n|^2','FontSize',12,'FontWeight','bold','Color','b')

    subplot(1,2,2);
    surf(abs(c2));
    colormap hsv;
    if M==16
        text(-5, 0.4, 'Fig a: Filtered Gabor Coefficient for ...
            log(nasdaq)','FontSize',16,'FontWeight','bold','Color','r')
    else
        text(-15, 0.4, 'Fig a: Filtered Gabor Coefficient for ...
            log(nasdaq)','FontSize',16,'FontWeight','bold','Color','r')
    end;
    xlabel('Time','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'xlabel');
    set(h,'rotation',30)
    ylabel('Frequency','FontSize',12,'FontWeight','bold','Color','b')
    h=get(gca,'ylabel');
    set(h,'Position',get(h,'Position') +[2 4 0])
    set(h,'rotation',140)
    zlabel('|C(m,n|^2','FontSize',12,'FontWeight','bold','Color','b')

end
```

```
function myreconsfromgabor()
%Program used to Gabor Coefficients for SP500 & NASDAQ
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: mygabor.m

```

```
    close all;
    clear all;
        [sp500,syear]=getData(1);
        sp500=log(sp500);
        [s2,s1]=hpfilter(sp500,1600);


    ∆m = 8;
    M=16;
   % M=50;
    ∆n=4;
    nn=32;
    %nn=100;
    thrcont=3;

    s1=s1';
    L=length(s1);
    t=1:L;
    N=L/2;
    nn2=nn/2;
    sigma=sqrt((∆m*L)/(∆n * 2 * pi));
    c=nthroot(pi*sigma*sigma,-4);
    h0 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
    h = @(ii) h0( mod(ii + N, L)-N);
    for m = 1:M
        for n = 1:nn2
            c1(m, n)= sum(s1.*h(mod(t - m*∆m,L)).*exp(-2*pi*i*∆n*n*t/L));
        end

    end


    [m,n]=size(c1);

    H=0.5;
    %thr=max(abs(c1));
    %spmask = (max(thr)-min(thr))/thrcont;
    spmask=mean(c1)+H*std(c1);
    for k = 1:m
        for j = 1:n
            if abs(c1(k,j)) < abs(spmask(k))
                sp500maskmatrix(k,j)=0;
            else
                sp500maskmatrix(k,j)=1;
            end;
        end;
    end;

        for t = 1:L
        temp=0;
            for m = 1:M
                for n = 1:nn2
                    temp= temp+c1(m,n).*h(mod(t - m*∆m,L)).*exp(2*pi*i*∆n*n*t/L);
                end
            end
            sg(t)=temp;
```

```

            end;
        sg=sg/(2*pi);
        c1=c1.*sp500maskmatrix;

            for t = 1:L
            temp=0;
                for m = 1:M
                    for n = 1:nn2
                        temp= temp+c1(m,n).*h(mod(t - m*Δm,M)).*exp(2*pi*i*Δn*n*t/L);
                    end
                end
                sg2(t)=temp;

            end;
            sg2=sg2/(2*pi);
           % plot(real(sg),'LineWidth',3);
           subplot(2,1,2);
            hold on;
            autocorr(s1,100);
            [c1,c2,c3]=autocorr(s1,100);
            hold on;
            p1=plot(c2,c1,'LineWidth',2,'Color','c');
            hold on;
            autocorr(real(sg2),100);
            [d1,d2,d3]=autocorr(real(sg2),100);
            hold on;
            p2=plot(d2,d1,'LineWidth',2,'Color','g');
            legend([p1,p2],'HP Original','HP Filtered');
             title('Original & Filtered HP Cycles');
            stdratio=std(real(sg2),1)/std(s1,1)
            vratio=var(real(sg2),1)/var(s1,1) * 100
            ccgo=corrcoef(real(sg2),s1)

            subplot(2,1,1);
            %figure;

            p1=plot(real(sg2),'LineWidth',.5,'DisplayName','Constructued signal');
            hold on;
            p2=plot(s1,'DisplayName','Original Signal');
            legend('show');
            ylabel('S(t)');
            xlabel('Time in year');
            title('log(SP500) original & Filtered HP Cycles');

            figure;
            clear all;

            [naq,nyear]=getData(2);
            naq=log(naq);
            [s2,s1]=hpfilter(naq,1600);
            Δm = 8;
      % M=16;
       M=50;
       Δn=4;
       %nn=32;
```

```
nn=100;
thrcont=3;

s1=s1';
L=length(s1);
t=1:L;
N=L/2;
nn2=nn/2;
sigma=sqrt((Δm*L)/(Δn * 2 * pi));
c=nthroot(pi*sigma*sigma,-4);
h0 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
h = @(ii) h0( mod(ii + N, L)-N);
for m = 1:M
    for n = 1:nn2
        c1(m, n)= sum(s1.*h(mod(t - m*Δm,L)).*exp(-2*pi*i*Δn*n*t/L));
    end

end


[m,n]=size(c1);

H=0.5;
%thr=max(abs(c1));
%spmask = (max(thr)-min(thr))/thrcont;
spmask=mean(c1)+H*std(c1);
for k = 1:m
    for j = 1:n
        if abs(c1(k,j)) < abs(spmask(k))
            sp500maskmatrix(k,j)=0;
        else
            sp500maskmatrix(k,j)=1;
        end;
    end;
end;

    for t = 1:L
    temp=0;
        for m = 1:M
            for n = 1:nn2
                temp= temp+c1(m,n).*h(mod(t - m*Δm,L)).*exp(2*pi*i*Δn*n*t/L);
            end
        end
        sg(t)=temp;

    end;
sg=sg/(2*pi);
c1=c1.*sp500maskmatrix;

    for t = 1:L
    temp=0;
        for m = 1:M
            for n = 1:nn2
                temp= temp+c1(m,n).*h(mod(t - m*Δm,M)).*exp(2*pi*i*Δn*n*t/L);
            end
        end
```

```
                 sg2(t)=temp;

              end;
              sg2=sg2/(2*pi);
             % plot(real(sg),'LineWidth',3);
             subplot(2,1,2);
              hold on;
              autocorr(s1,100);
              [c1,c2,c3]=autocorr(s1,100);
              hold on;
              p1=plot(c2,c1,'LineWidth',2,'Color','c');
              hold on;
              autocorr(real(sg2),100);
              [d1,d2,d3]=autocorr(real(sg2),100);
              hold on;
              p2=plot(d2,d1,'LineWidth',2,'Color','g');
              legend([p1,p2],'HP Original','HP Filtered');
              title('Original & Filtered HP Cycles');
              stdratio=std(real(sg2),1)/std(s1,1)
              vratio=var(real(sg2),1)/var(s1,1) * 100
              ccgo=corrcoef(real(sg2),s1)

              subplot(2,1,1);
             %figure;

              p1=plot(real(sg2),'LineWidth',.5,'DisplayName','Constructued signal');
              hold on;
              p2=plot(s1,'DisplayName','Original Signal');
              legend('show');
              ylabel('S(t)');
              xlabel('Time in year');
              title('log(NASDAQ) original & Filtered HP Cycles');
end
```

```
%Program used to Gabor Coefficients for SP500 & NASDAQ
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: mygabor.m

       close all;
       clear all;
           [sp500,syear]=getData(1);
           sp500=log(sp500);
           [s2,s1]=hpfilter(sp500,1600);


       ∆m = 12;
       M=16;
       %M=50;
       ∆n=4;
       nn=32;
       %nn=100;
```

```
        thrcont=1;
        s1=s1';
        L=length(s1);
        t=1:L;
        N=L/2;
        nn2=nn/2;
        sigma=sqrt((Δm*L)/(Δn * 2 * pi));
        c=nthroot(pi*sigma*sigma,-4);
        h0 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
        h = @(ii) h0( mod(ii + N, L)-N);
        for m = 1:M
            for n = 1:nn2
                c1(m, n)= sum(s1.*h(mod(t - m*Δm,L)).*exp(-2*pi*i*Δn*n*t/L));
            end

        end
        [m,n]=size(c1);
        H=0.5;
        %thr=max(abs(c1));
        %spmask = (max(thr)-min(thr))/thrcont;
        spmask=mean(c1)+H*std(c1);
        for k = 1:m
            for j = 1:n
           if abs(c1(k,j)) < abs(spmask(k))
         %       if abs(c1(k,j)) < spmask

                sp500maskmatrix(k,j)=0;
                else
                    sp500maskmatrix(k,j)=1;
                end;
            end;
        end;

            for t = 1:L
            temp=0;
                for m = 1:M
                    for n = 1:nn2
                        temp= temp+c1(m,n).*h(mod(t - m*Δm,L)).*exp(2*pi*i*Δn*n*t/L);
                    end
                end
                sg(t)=temp;

            end;
        sg=sg/(2*pi);
        c1=c1.*sp500maskmatrix;
        for t = 1:L
            temp=0;
                for m = 1:M
                    for n = 1:nn2
                        temp= temp+c1(m,n).*h(mod(t - m*Δm,M)).*exp(2*pi*i*Δn*n*t/L);
                    end
                end
                sg2(t)=temp;
         end;
           Δ=60;
          % subplot(2,1,1);
```

```
            % x=real(s1(1:length(s1)-Δ));
            % y=real(s1(Δ+1:length(s1)));
            % plot(x,y);
            % subplot(2,1,2);
              figure;
              x=real(sg2(1:length(sg2)-Δ));
              y=real(sg2(Δ+1:length(sg2)));
              plot(x,y,'LineWidth',.7,'Color','r');
              xlabel('x(t)','FontSize',12,'FontWeight','bold','Color','b')
              ylabel('x(t+T)','FontSize',12,'FontWeight','bold','Color','b');
              title('SP500 Filtered HP cycles','FontSize',12,'FontWeight','bold','Color','b');


              sum(sum(sp500maskmatrix))
              m*n


              clear all;
              [nas,syear]=getData(2);
              nas=log(nas);
              [s2,s1]=hpfilter(nas,1600);


        Δm = 8;
        M=16;
       % M=50;
        Δn=4;
        nn=32;
        %nn=100;
        thrcont=1;
        s1=s1';
        L=length(s1);
        t=1:L;
        N=L/2;
        nn2=nn/2;
        sigma=sqrt((Δm*L)/(Δn * 2 * pi));
        c=nthroot(pi*sigma*sigma,-4);
        h0 = @(b) c*exp(-((b.*b)/(2*sigma*sigma)));
        h = @(ii) h0( mod(ii + N, L)-N);
        for m = 1:M
            for n = 1:nn2
                c1(m, n)= sum(s1.*h(mod(t - m*Δm,L)).*exp(-2*pi*i*Δn*n*t/L));
            end

        end
        [m,n]=size(c1);
        H=0.5;
        %thr=max(abs(c1));
        %spmask = (max(thr)-min(thr))/thrcont;
        spmask=mean(c1)+H*std(c1);
        for k = 1:m
            for j = 1:n
           if abs(c1(k,j)) < abs(spmask(k))
         %       if abs(c1(k,j)) < spmask

```

```
                sp500maskmatrix(k,j)=0;
                else
                    sp500maskmatrix(k,j)=1;
                end;
            end;
        end;

            for t = 1:L
            temp=0;
                for m = 1:M
                    for n = 1:nn2
                        temp= temp+c1(m,n).*h(mod(t - m*Δm,L)).*exp(2*pi*i*Δn*n*t/L);
                    end
                end
                sg(t)=temp;

            end;
        sg=sg/(2*pi);
        c1=c1.*sp500maskmatrix;
        for t = 1:L
            temp=0;
                for m = 1:M
                    for n = 1:nn2
                        temp= temp+c1(m,n).*h(mod(t - m*Δm,M)).*exp(2*pi*i*Δn*n*t/L);
                    end
                end
                sg2(t)=temp;
         end;
            Δ=60;
            figure;
            x=real(sg2(1:length(sg2)-Δ));
            y=real(sg2(Δ+1:length(sg2)));
            plot(x,y,'LineWidth',.7,'Color','b');
            xlabel('x(t)','FontSize',12,'FontWeight','bold','Color','r')
            ylabel('x(t+T)','FontSize',12,'FontWeight','bold','Color','r');
            title('NASDAQ Filtered HP cycles','FontSize',12,'FontWeight','bold','Color','r');

            sum(sum(sp500maskmatrix))
            m*n
```

APPENDIX C

Mathematical Proof

C.1 Gabor Elementary function

$$\psi(t) = \underbrace{e^{-\alpha^2(t-t_0)^2}}_{v}\overbrace{e^{j2\pi f_0 t+\phi}}^{w} \tag{C.1}$$

v represents the probability function and w represents simple harmonic oscillator. $\Psi(f)$ is the GEF in the frequency domain. The GEF in the frequency domain is attained by taking the Fourier transform of the GEF.

$$\Psi(f) = \int_{-\infty}^{\infty} \psi(t)e^{-j2\pi ft}dt; \Psi(f) = \int_{-\infty}^{\infty} e^{-\alpha^2(t-t_0)^2}e^{j2\pi f_0 t+\phi}e^{-j2\pi ft}dt$$

$$\Psi(f) = \int_{-\infty}^{\infty} e^{-\alpha^2(t-t_0)^2}e^{j2\pi t(f_0-f)+\phi}dt$$

$$\Psi(f) = e^{\phi}\int_{-\infty}^{\infty} e^{-\alpha^2(t-t_0)^2}e^{j2\pi t(f_0-f)}dt$$

when t_0 is 0, then

$$\Psi(f) = e^{\phi} \int_{-\infty}^{\infty} e^{-\alpha^2 t^2} e^{j2\pi t(f_0 - f)} dt \tag{C.2}$$

This is of the form.

$$\int_{-\infty}^{\infty} e^{2bx - ax^2} dx = \sqrt{\frac{\pi}{a}} e^{\frac{b^2}{a}}$$

where $b = j\pi(f_0 - f)$ and $a = \alpha^2$

$$\Psi(f) = \sqrt{\frac{\pi}{\alpha^2}} e^{\frac{(j\pi(f_0 - f))^2}{\alpha^2}} e^{\phi}$$

$$\Psi(f) = \sqrt{\frac{\pi}{\alpha^2}} e^{-(\frac{\pi}{\alpha})^2 (f_0 - f)^2 + \phi}$$

α is connecting the GEF between time and frequency domain. $\psi(t)$ and $\Psi(f)$ occupies the minimum uncertainty in time and frequency domain.

C.1.1 Proof: GEF has minimum uncertainty in the time-frequency domain

I believe, we will better understand physical or mathematical concept by performing a step wise derivation. Let me do a step wise derivation to prove that GEF has a minimum uncertainty for a special case. Let me simplify the GEF by taking GEF at zero frequency, $t_0 = 0$ and $\phi = 0$,the Gabor elementary function and Fourier transform of GEF are given by

$$\psi(t) = e^{-\alpha^2 t^2}$$

$$\Psi(f) = \sqrt{\frac{\pi}{\alpha^2}} e^{-(\frac{\pi}{\alpha})^2 f^2}$$

Effective duration Δt is given by:

$$\Delta t = \sqrt{\frac{\int_{-\infty}^{\infty} e^{-\alpha^2 t^2} t^2 e^{-\alpha^2 t^2} dt}{\int_{-\infty}^{\infty} e^{-\alpha^2 t^2} e^{-\alpha^2 t^2} dt}};$$

Let me take the denominator first

$$\int\limits_{-\infty}^{\infty} e^{-\alpha^2 t^2} e^{-\alpha^2 t^2} dt = \int\limits_{-\infty}^{\infty} e^{-2\alpha^2 t^2} dt$$

The above equation is of the form and it only applies when $a > 0$

$$\int\limits_{-\infty}^{\infty} e^{-ax^2} dx = \sqrt{\frac{\pi}{a}}$$

where $a = 2\alpha^2$

$$\int\limits_{-\infty}^{\infty} e^{-2\alpha^2 t^2} dt = \sqrt{\frac{\pi}{2\alpha^2}}$$

Let me take the numerator now,

$$\int\limits_{-\infty}^{\infty} e^{-\alpha^2 t^2} t^2 e^{-\alpha^2 t^2} dt = \int\limits_{-\infty}^{\infty} t^2 e^{-2\alpha^2 t^2} dt$$

The above equation is of the form.

$$\int\limits_{-\infty}^{\infty} x^2 e^{-ax^2} dx = \frac{1}{2}\sqrt{\frac{\pi}{a^3}}$$

where $a = 2\alpha^2$

$$\int\limits_{-\infty}^{\infty} t^2 e^{-2\alpha^2 t^2} dt = \frac{1}{2}\sqrt{\frac{\pi}{(2\alpha^2)^3}} = \frac{\sqrt{\pi}}{4\sqrt{2}\alpha^3}$$

Let me apply both numerator and denominator value to get the effective duration Δt

$$\Delta t = \sqrt{\frac{\frac{\sqrt{\pi}}{4\sqrt{2}\alpha^3}}{\sqrt{\frac{\pi}{2\alpha^2}}}};$$

Straight forward steps to simply the value of Δt

$$\Delta t = \sqrt{\frac{\sqrt{\pi}}{4\sqrt{2}\alpha^3}\sqrt{\frac{2\alpha^2}{\pi}}};$$

$$\Delta t = \sqrt{\frac{\sqrt{\pi}}{4\sqrt{2}\alpha^3}\sqrt{\frac{2\alpha^2}{\pi}}};$$

$$\Delta t = \sqrt{\frac{1}{4\alpha^2}};$$

$$\Delta t = \frac{1}{2\alpha} \tag{C.3}$$

Let me do the similar steps to calculate the effective frequency Δf. The frequency representation of the GEF is given by,

$$\Psi(f) = \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2}$$

Effective frequency Δf is given by,

$$\Delta f = \sqrt{\frac{\int_{-\infty}^{\infty} \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2} f^2 \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2} df}{\int_{-\infty}^{\infty} \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2} \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2} df}};$$

Let me take the denominator first.

$$\int_{-\infty}^{\infty} \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2} \sqrt{\frac{\pi}{\alpha^2}}e^{-(\frac{\pi}{\alpha})^2 f^2} df = \frac{\pi}{\alpha^2}\int_{-\infty}^{\infty} e^{-2(\frac{\pi}{\alpha})^2 f^2} df$$

The above equation is of the form and it only applies when $a > 0$

$$\int_{-\infty}^{\infty} e^{-ax^2} dx = \sqrt{\frac{\pi}{a}}$$

where $a = 2(\frac{\pi}{\alpha})^2$

$$\frac{\pi}{\alpha^2} \int_{-\infty}^{\infty} e^{-2(\frac{\pi}{\alpha})^2 f^2} df = \frac{\pi}{\alpha^2} \sqrt{\frac{\pi}{2(\frac{\pi}{\alpha})^2}}$$

Let $\beta = \frac{\pi}{\alpha}$

$$\frac{\pi}{\alpha^2} \int_{-\infty}^{\infty} e^{-2(\frac{\pi}{\alpha})^2 f^2} df = \frac{\beta}{\alpha} \sqrt{\frac{\pi}{2\beta^2}} = \frac{1}{\alpha} \sqrt{\frac{\pi}{2}}$$

Let me take the numerator now.

$$\int_{-\infty}^{\infty} \sqrt{\frac{\pi}{\alpha^2}} e^{-(\frac{\pi}{\alpha})^2 f^2} f^2 \sqrt{\frac{\pi}{\alpha^2}} e^{-(\frac{\pi}{\alpha})^2 f^2} df$$

$$= \frac{\pi}{\alpha^2} \int_{-\infty}^{\infty} f^2 e^{-2(\frac{\pi}{\alpha})^2 f^2} df$$

Substitute β in above equation.

$$= \frac{\beta}{\alpha} \int_{-\infty}^{\infty} f^2 e^{-2\beta^2 f^2} df$$

The above equation is of the form.

$$\int_{-\infty}^{\infty} x^2 e^{-ax^2} dx = \frac{1}{2} \sqrt{\frac{\pi}{a^3}}$$

where $a = 2\beta^2$

$$= \frac{\beta}{\alpha} \frac{1}{2} \sqrt{\frac{\pi}{8\beta^6}}$$

$$= \frac{\beta}{\alpha} \frac{\sqrt{\pi}}{4\sqrt{2}\beta^3}$$

Substitute the value of β

$$= \frac{1}{\alpha} \frac{\sqrt{\pi}}{4\sqrt{2}\beta^2} = \frac{1}{\alpha} \frac{\sqrt{\pi}\alpha^2}{4\sqrt{2}\pi^2} = \frac{\sqrt{\pi}\alpha}{4\sqrt{2}\pi^2}$$

Apply the value of numerator and denominator of Δf

$$\Delta f = \sqrt{\frac{\frac{\sqrt{\pi}\alpha}{4\sqrt{2}\pi^2}}{\frac{1}{\alpha}\sqrt{\frac{\pi}{2}}}} = \sqrt{\frac{\sqrt{\pi}\alpha^2}{4\sqrt{2}\pi^2}\sqrt{\frac{2}{\pi}}}$$

Step wise simplification steps to get the value of Δf

$$\Delta f = \sqrt{\frac{\alpha^2}{4\pi^2}}$$

$$\Delta f = \frac{\alpha}{2\pi} \tag{C.4}$$

Apply both the value of Δf and Δt from equation (8) and equation (9)

$$\Delta t \Delta f = \frac{\alpha}{2\pi} \frac{1}{2\alpha}$$

$$\boxed{\Delta t \Delta f = \frac{1}{4\pi}}$$

Hence the proof.

BIBLIOGRAPHY