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

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

CHAPTER II

Gabor Transformation

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 and spatial-frequency variables. WD belongs to a large class of bilinear distributions known as the Cohen's class, in which each member can be obtained choosing a different kernel in the generalized bilinear distribution definition.

Let's 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

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

where F^* is the complex conjugate of F. Connecting 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 the WD, the interference and inversion properties are most relevant for the current study. The WD computation introduces a spurious "auto terms" due to its intrinsic bi-linearity. The WD of sum of two signals $f(t)+f'(t)$ is given by

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

Inversion of the original signal $f(t)$ from the Wigner Distribution is given by

$$f(t+\frac{\tau}{2})f^*(t-\frac{\tau}{2})=\int\limits_{-\infty}^{\infty}W_f(t,\omega)e^{j\omega\tau}d\omega \tag{3.6}$$

Let $t=\frac{\tau}{2}$ and then setting $\tau=t$,we have

$$f(t)f^*(0)=\int\limits_{-\infty}^{\infty}W_f(\frac{t}{2},\omega)e^{j\omega t}d\omega \tag{3.7}$$

$$f(t)=\frac{1}{f^*(0)}\int\limits_{-\infty}^{\infty}W_f(\frac{t}{2},\omega)e^{j\omega t}d\omega \tag{3.8}$$

3.2 Wigner-Ville Distribution

The Wigner-Ville distribution (WVD)is defined for a signal $f(t)$ as follows:

$$W_f(t,\omega) = \int_{-\infty}^{\infty} z(t+\frac{\tau}{2})z^*(t-\frac{\tau}{2})e^{-j\omega\tau}d\tau \tag{3.9}$$

where $z(t)$ is the analytic associative of $f(t)$. The Wigner distribution is simply defined when the real signal $f(t)$ is used instead of the analytic one $z(t)$. A signal $f(t)$ is said to be analytical if and only if

$$F(\omega) = 0 \tag{3.10}$$

for all $\omega < 0$,where $F(\omega)$ is Fourier transform of $f(t)$. In other words, an analytical signal contains no negative frequencies. It may have a spectral component at zero frequency.

3.3 Discrete Wigner Distribution

One of the main disadvantages of the discrete definition is that not all the properties of the continuous WD are preserved by discretization due to aliasing effects. Several alternative 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

$$W_f(n,m) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} f(n+k)f^*(n-k)e^{-j(\frac{2\pi mk}{N+1})} \tag{3.11}$$

where n and m 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.

Inversion property is the ability to extract the time domain signal from distribution,up to a constant phase factor and proposed distribution satisfies this property as

$$f(n+k)f^*(n-k) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} W_f(n,m)e^{j(\frac{2\pi mk}{N+1})} \tag{3.12}$$

In the case of discrete signals, inserting $k = n$ in the above equation allows one to write

$$f(2n)f^*(0) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} W_f(n,m)e^{j(\frac{2\pi mn}{N+1})} \tag{3.13}$$

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

$$f(2n-1)f^*(1) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} W_f(n,m)e^{j(\frac{2\pi mn}{N+1})} \tag{3.14}$$

leads to the recovering of the odd samples.

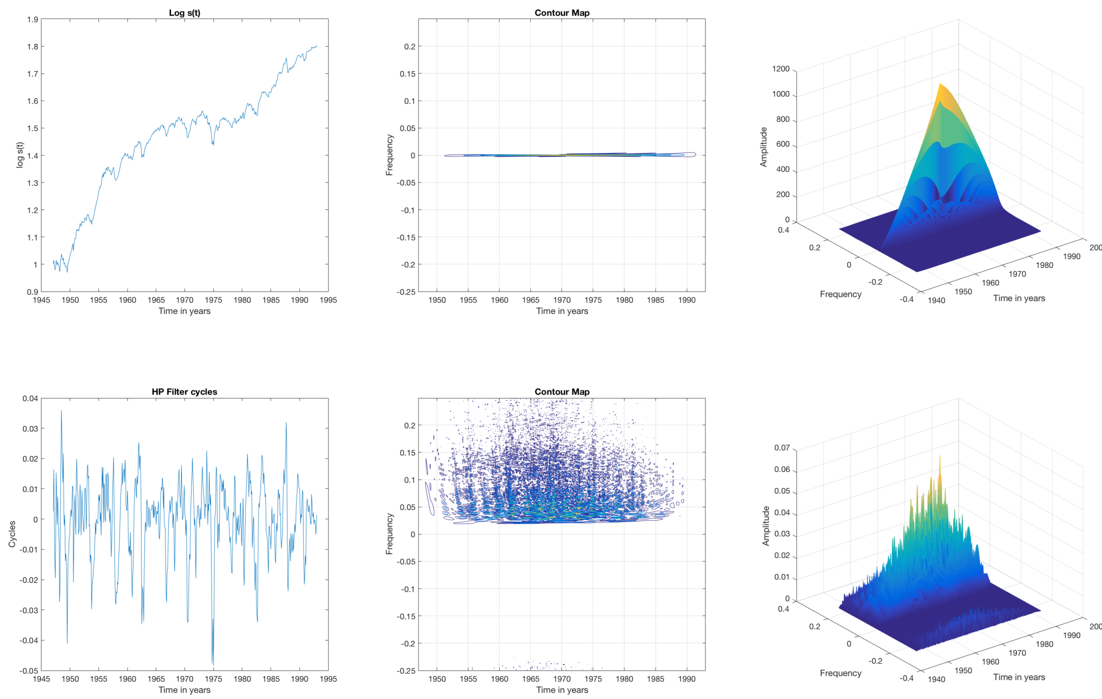

Figure 3.1: Wigner Distribution transform for log(s); where s denotes the sp500. Top row is the log(s), Wigner Distribution transform presentation in the contour and three dimensional mesh. Bottom row is the HP filter cycle of log(s) with $\lambda = 1600$ and its Wigner Distribution transform. The graph was created using Matlab code mywvd.m and it is attached in the Appendix B

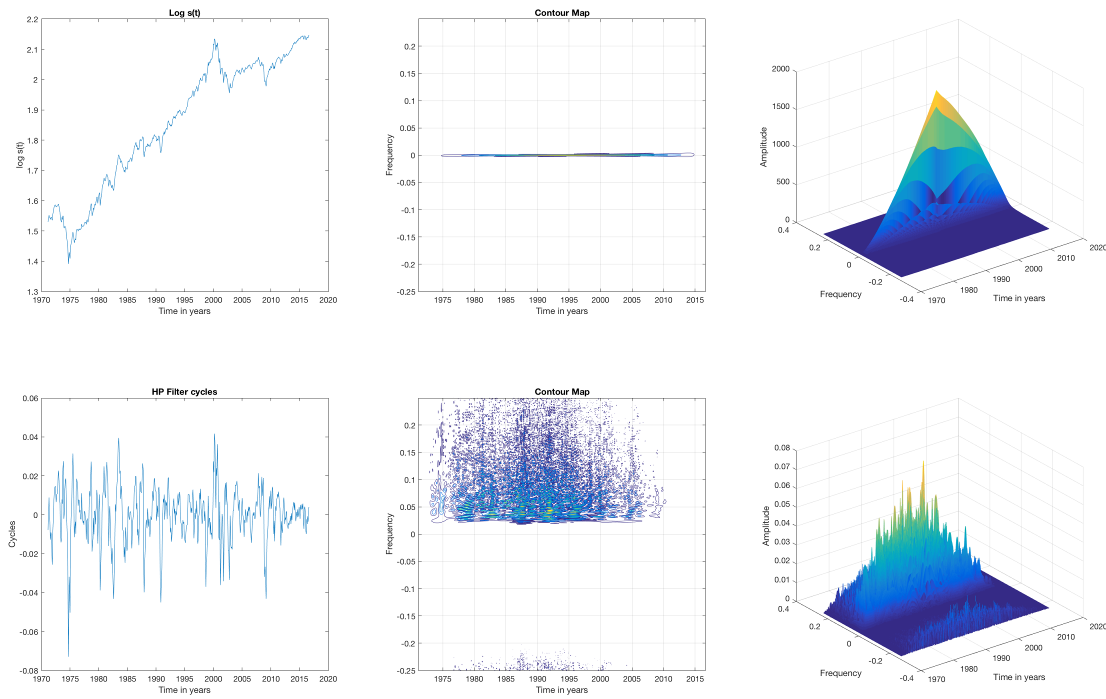

Figure 3.2: Wigner Distribution transform for log(s); where s denotes the NASDAQ index. Top row is the log(s), Wigner Distribution transform presentation in the contour and three dimensional mesh. Bottom row is the HP filter cycle of log(s) with $\lambda = 1600$ and its Wigner Distribution transform. 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 is localized.

Time Frequency Distribution Series (TFDS) was introduced by Chen and Qian [2] as the 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 which is defined as follows.

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

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. Let

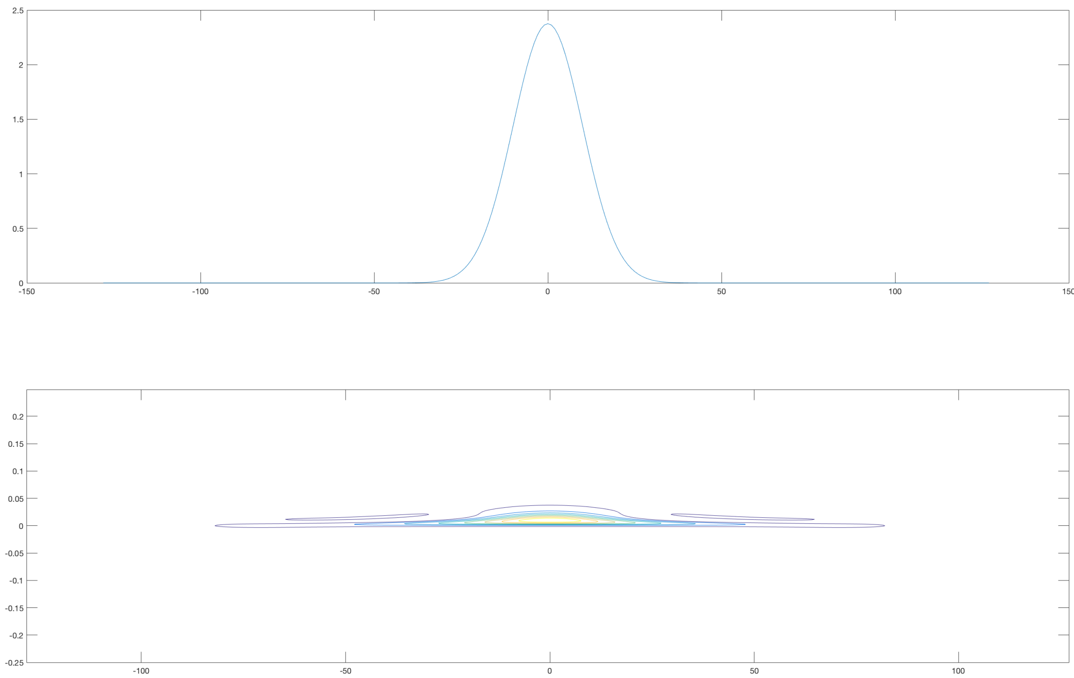

Figure 4.1: Fig in the top represents the Gaussian function $g(t)$ as defined above with $\sigma = 0.1$. Fig in the bottom represents the Wigner Ville Distribution WVD_g as defined above. The graph is created using the mywvdgauss.m program attached in the Appendix. WVD values are created by using the HOSA (Higher Order Spectral Analysis) Matlab toolbox.

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

where, Ω and T are the frequency and time sampling steps respectively. The WVD of $h(t)$ is given by,

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

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

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

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

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 the cross term. When a signal $s(t)$ can be decomposed as a linear combination of some elementary functions $h(t)$ then the cross-terms can be controlled.

Recall from the previous chapter on the Gabor Expansion,for 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

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

Using the Wigner-Ville distribution of $s(t)$ 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 $n = n'$,

$$C_{m,n}C^*_{m',n'}WVD_{h,h'}(t,\omega) = 2|C_{m,n}|^2 e^{-[\frac{(t-mT)^2}{\sigma^2}+\sigma^2(\omega-n\Omega)^2)]} \tag{4.15}$$

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

$$C_{m,n}C^*_{m',n'}WVD_{h,h'}(t,\omega) = C_{m,n}C^*_{m',n'}e^{j\omega_d t_\mu}H(t-t_\mu,\omega-\omega_\mu) \tag{4.16}$$

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

$$TFDS_D(t,\omega) = \sum_{d=0}^{D} P_d(t,\omega) \tag{4.17}$$

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

$$P_d(t,\omega) = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C^*_{m',n'} WVD_{h,h'}(t,\omega) \tag{4.18}$$

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

$$P_d(t,\omega) = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C^*_{m',n'} e^{j\omega_d t_\mu} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[t_d\omega - \omega_d t]} \tag{4.19}$$

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

$$P_d(t,\omega) = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C^*_{m',n'} e^{j\frac{n+n'}{2}\Omega(m-m')T} 2e^{-[\frac{t^2}{\sigma^2}+(\sigma\omega)^2]} e^{-j[(m-m')T\omega-(n-n')\Omega t]} \tag{4.20}$$

Both the SP500 and NASDAQ indices are discrete signals used for analysis. The P_d is further simplified for programming convenience.

The discrete Time Frequency Distribution Series is defined as follows.

$$DTFDS_D[i,k] = TFDS_d(t,\omega)|_{t=i\Delta t,\omega=\frac{2\pi k}{L\Delta t}} \tag{4.21}$$

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

$$TFDS_d(i,k) = \sum_{d=0}^{D} P_d[i,k] \tag{4.22}$$

where

$$P_d[i,k] = \sum_{|m-m'|+|n-n'|=d} C_{m,n} C_{m'n'} WVD_{h,h'}[i,k] \tag{4.23}$$

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.

$$WVD_s[i,k] = WVD_s(t,\omega)|_{t=i\Delta t, \omega=\frac{2\pi k}{L\Delta t}} \tag{4.24}$$

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

$$WVD_{h,h'}[i,k] = 2e^{-\sigma(i-\frac{m+m'}{2}\Delta M)^2 - \frac{1}{\sigma}(k-\frac{n+n'}{2}\Delta N)^2} e^{j\frac{2\pi}{L}[(m-m')\Delta Mk + (n-n')\Delta Ni - \frac{n+n'}{2}\Delta N(m-m')\Delta M]} \tag{4.25}$$

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

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 mywvdgauss()

%Program used to Wigner Ville Distribution for Gaussian function
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: mywvdgauss.m

close all;
clear all;
t=-128:127;
sigma=.1;
coef=nthroot(pi*sigma*sigma,-4);
expo=(t.*t)/2*sigma*sigma;
g=coef*exp(-expo);
subplot(2,1,1);
plot(t,g);
[wd,freq]=wig2(g);
subplot(2,1,2)
contour(t,freq,wd',8)
```

```
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*Δm,L)).*exp(-2*pi*i*Δn*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: myfiltgabor.m
close all;
clear all;
[sp500,syear]=getData(1);
[naq,nyear]=getData(2);
    ∆m = 8;
    M=16;
    %M=50;
    ∆n=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((∆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
    [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: myreconsfromgabor.m

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


        Δm = 8;
        M=17;
       % M=50;
        Δn=4;
        nn=84;
        %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(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: myAttractor.m

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


        Δm = 12;
        M=17;
        %M=50;
        Δn=4;
        nn=84;
        %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
```

```
function myFDfilter()
%Program used to Gabor Coefficients for SP500 & NASDAQ
% Praba Siva
% praba@umich.edu
% @prabasiva
% Filename: myFDfilter.m
close all;
clear all;
[sp500,syear]=getData(1);
process(sp500,1);
[naq,nyear]=getData(2);
process(naq,2);


function process(sp500,flag)

 % First differening
 % Graph looks identifical as shown in the P.Chen's paper
 % P.Chen's paper says it is log of sp500 data but he used sp500
 % The graph in his paper shows it all.

    len=size(sp500);
    t1=sp500(1:len-1);
    t2=sp500(2:len);
    fd=t2-t1;
    lenfd=size(fd);
    Δ=40;
    x=fd(1:lenfd-Δ);
    y=fd(Δ+1:lenfd);
    figure;
    subplot(2,1,1);
    scatter(x,y,'b');
    xlabel('X(t)','Fontsize',20);
    ylabel('X(t+T)', 'Fontsize', 20);
    if flag ==1
        title('FD Series for SP500 Index','Fontsize',20);
    else
        title('FD Series for NASDAQ Index','Fontsize',20);
    end;
    y=fd;
    n=size(y);
    % C and Phi values are given in the P.Chen's paper.

    c = 0.006*0.002;
    phi = [0.000265*0.043, -0.81*0.043];
```

```

    % White noise created based on the standard deviation value
    noise = 0.033*randn(n(1),1);

    for t=3:n
        x(t-2) = c + phi(1)*y(t-1) + phi(2)*y(t-2) +noise(t-2);
    end
    lenfd=size(x);
    Δ=5;
    x1=x(1:lenfd-Δ);
    y1=x(Δ+1:lenfd);
    subplot(2,1,2);
    scatter(x1,y1,'r');
    xlabel('X(I)','Fontsize',20);
    ylabel('X(I+T)', 'Fontsize', 20);
    if flag ==1
         title('AR(2)for FDSeries of SP500 index','Fontsize',20);
    else
        title('AR(2)for FDSeries of NASDAQ index','Fontsize',20);
    end;
end
end
```

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.

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

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

BIBLIOGRAPHY

[1] D. Gabor. Theory of Communication. *J. IEE*, 93(26):429–457, November 1946.

[2] Ping Chen. A Random-Walk or Color-Chaos on the Stock Market? - Time-Frequency Analysis of S&P Indexes. *Studies in Nonlinear Dynamics & Econometrics*, 1(A9):87–103, July 1996.

[3] Morten O. Ravn and Harald Uhlig. On Adjusting the Hodrick-Prescott filter for the frequency of observation. *Notes*, 1996.