Saha Institute Of Nuclear Physics

Study and Analysis of Non-Magnetized Plasma Systems

Utsab Chowdhury

Supervisor: Prof. A.N Sekar Iyengar

A report submitted in partial fulfilment of the requirements
of the summer project at SINP

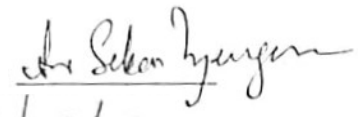
in the

Department of Plasma Physics

July 9, 2019

Certificate of Recommendation

This is to certify that Utsab Chowdhury, has been involved in his project work involving non-linear analysis of plasma systems under the direct supervision and guidance of Prof.A.N.Sekar Iyengar. I am satisfied with his work, which is being presented for the partial fulfillment of the requirements the summer project at Saha Institute of Nuclear Physics

Signature 

Date 16/07/19

Acknowledegement

The work presented in this report has been carried out at Saha Institute of Nuclear Physics, Kolkata during August-December, 2018. It is my immense pleasure to express my sincere gratitude to all the people who have made this dissertation project possible by their invaluable contribution and support. Would it not have been for all these people, this project work would never have been possible to carry out.

First of all, I would like to express my sincere gratitude and heartfelt thanks to my supervisers, Prof. M. S. Janaki and Prof. A. N. Sekar Iyenger of Plasma Physics Division, Saha Institute of Nuclear Physics for their careful guidance, encouragement and moral support throughout this project. Without which, I would not have been able to complete my work this semester. Prof. A. N. S. Iyenger gave me the freedom to work independently on the glow discharge plasma experiment, and he always encouraged to tackle the problem faced with simple solutions. He also created a friendly environment during our discussions and encouraged to ask questions regarding every aspect of the experiment.

Contents

1 Introduction to Plasma **1**
- 1.1 What is Plasma? . . . 1
- 1.2 Definition of Plasma . . . 2
  - 1.2.1 Debye shielding and Plasma frequency . . . 2
  - 1.2.2 Plasma characterization . . . 2
- 1.3 Waves and oscillations in Plasma . . . 3
- 1.4 Plasma and Non linear dynamics . . . 3

2 The Glow Discharge Plasma Experiment **5**
- 2.1 Introduction . . . 5
- 2.2 Experimental Setup: Glow Discharge Plasma . . . 5
- 2.3 Verification of Paschen Curve in Glow Discharge Plasma . . . 7
- 2.4 Langmuir Probe . . . 9
  - 2.4.1 Electron Saturation Region . . . 10
  - 2.4.2 Electron Retard Region . . . 11
  - 2.4.3 Ion Saturation Region . . . 11

3 Data Analysis Techniques: Time Series Analysis **12**
- 3.1 Introduction . . . 12
- 3.2 Power Spectral Method . . . 12
  - 3.2.1 The Fourier Theorem . . . 12
  - 3.2.2 The Fourier Transform . . . 13
  - 3.2.3 Nyquist Critical Frequency . . . 13
  - 3.2.4 Fast Fourier Transform . . . 13
- 3.3 Phase Space Reconstruction . . . 18
  - 3.3.1 Optimal Embedding Dimension . . . 21
- 3.4 Recurrence Plots . . . 25
- 3.5 Filtering . . . 28
  - 3.5.1 Singular Value Decomposition (SVD) . . . 28

4 Conclusions **53**

Appendices **55**

A MATLAB Codes **56**
A.1 1. Fast Fourier Transform (FFT) of the Time series 56
A.2 Phase Portrait 59
A.2.1 Phase Portrait of various signals 59
A.2.2 Optimal Embedding Dimension 63
A.3 Recurrence Plots 68
A.4 Filtering 70
A.4.1 Singular Value Decomposition(SVD) 70
A.4.2 Main Code 70
A.4.3 SVD function 74

List of Figures

2.1 Schematic Diagram of the cylindrical electrode system of the Glow-Discharge plasma 6
2.2 Schematic Diagram of the Vacuum chamber setup of the Glow-Discharge plasma; 1. The cylindrical cathode, 2. The central anode rod, 3. The Langmuir probe, 4. The grid probe, 5. Connection to oil rotary vacuum pump, 6. Pirani pressure gauge probe and its connection, 7. Argon gas inlet pipe and needle valve, 8. Argon gas cylinder. 7
2.3 V_{br} vs p for the experiment of verification of Paschen curve 8
2.4 A typical Langmuir Probes I-V characteristic curve 10

3.1 (a) time domain and (b) frequency domain plot of a sample sine wave of frequency 1000 Hz 15
3.2 (a) shows sine wave superposed with two sine waves of frequency 1000Hz and 2000Hz respectively (b) After obtaining FFT we can see the frequencies of the mixture 16
3.3 ((a) shows the sawtooth wave signal (c) After obtaining FFT we can see the frequencies of the mixture 17
3.4 ((a) shows the time series data in time domain and (b) shows the FFT of the same time series 18
3.5 (a) A Periodic sinusoidal signal (b) Phase portrait of the periodic signal . . . 19
3.6 (a) A Quasi-periodic signal (b) Phase portrait of the quasi-periodic signal . . 20
3.7 (a) A noisy periodic signal (b) Phase portrait of the noisy periodic signal . . 20
3.8 (a) A non-linear signal (b) Phase portrait of the non-linear signal 21
3.9 (a) A non-linear signal from the plasma system (b) Phase portrait of the non-linear signal 23
3.10 Variation of E1 and E2 with respect to embedding dimension of the given time series data above, x-axis is (d) and y-axis is E1 E2 23
3.11 (a) A non-linear signal from the plasma system (b) Phase portrait of the non-linear signal 24
3.12 Variation of E1 and E2 with respect to embedding dimension of the given time series data above, x-axis is (d) and y-axis is E1 E2 24
3.13 Recurrence plot of the time series data non-linear signal in Figure 3.9 26
3.14 Recurrence plot of the time series data non-linear signal in Figure 3.11 27
3.15 Original Time Series of the signal from the plasma system 30
3.16 FFT of the unfiltered signal from the plasma system 31

3.17 log-log FFT of the unfiltered signal from the plasma system 32
3.18 Phase portrait of the unfiltered signal from the plasma system 33
3.19 FFT after SVD analysis of the signal from the plasma system 34
3.20 log-log FFT after SVD analysis of the signal from the plasma system 35
3.21 Reconstructed time series after applying butterworth filter over the SVD analysed time series 36
3.22 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 37
3.23 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 38
3.24 Phase portrait of reconstructed time series after applying butterworth filter over the SVD analysed time series 39
3.25 FFT after SVD analysis of the signal from the plasma system 40
3.26 log-log FFT after SVD analysis of the signal from the plasma system 41
3.27 Reconstructed time series after applying butterworth filter over the SVD analysed time series 42
3.28 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 43
3.29 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 44
3.30 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 45
3.31 FFT after SVD analysis of the signal from the plasma system 46
3.32 log-log FFT after SVD analysis of the signal from the plasma system 47
3.33 Reconstructed time series after applying butterworth filter over the SVD analysed time series 48
3.34 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 49
3.35 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 50
3.36 FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series 51

Chapter 1

Introduction to Plasma

Under neutral conditions, three stares of matter can be observed in the universe: solid state, liquid state and the gaseous state. However, a fourth state of matter can also be found, known as the Plasma state. Plasma is an semiionized gas, i.e in this state of matter, in addition to atoms and molecules, it also contains ions (both positive and negative) and electrons. Most of the observable universe is made up of plasma. For example, stars along with their corona and wind, the magneto-spheres of earth (in presence of the solar wind) and other planets, the tails of comets, nebulae, interstellar and inter-galactic media, accretion disks around black holes are all made up of plasma. There are also plasma here on Earth, ranging from the inside of a nuclear fusion reactor to a candle flame. The ionosphere, a layer of Earths atmosphere that extends from 85 to 600 km, contains plasma. It is a result of the ionization of oxygen, nitrogen and nitric oxide by ultraviolet and Xray radiation coming from the Sun. Another example of a natural plasma observed on the Earth is the lightning, made by the electrical discharge between oppositely charged regions, i.e the cloud and the ground. The light produced during a lightning is a result of the ionization of the air. In made made systems also plasmas can be seen, such as in the jets of rockets, in the fireball of the nuclear explosion, on the surface of space vehicles during going in or out of earth atmosphere, in the welding guns, in tube-lights and in many more systems.

1.1 What is Plasma?

Under neutral conditions, three stares of matter can be observed in the universe: solid state, liquid state and the gaseous state. However, a fourth state of matter can also be found, known as the Plasma state. Plasma is an semiionized gas, i.e in this state of matter, in addition to atoms and molecules, it also contains ions (both positive and negative) and electrons. Most of the observable universe is made up of plasma. For example, stars along with their corona and wind, the magneto-spheres of earth (in presence of the solar wind) and other planets, the tails of comets, nebulae, interstellar and inter-galactic media, accretion disks around black holes are all made up of plasma. There are also plasma here on Earth, ranging from the inside of a nuclear fusion reactor to a candle flame. The ionosphere, a layer of Earths atmosphere that extends from 85 to 600 km, contains plasma. It is a result of the ionization of oxygen, nitrogen and nitric oxide by ultraviolet and Xray radiation coming from the Sun. Another example of a natural plasma observed on the Earth is the lightning, made

by the electrical discharge between oppositely charged regions, i.e the cloud and the ground. The light produced during a lightning is a result of the ionization of the air. In made made systems also plasmas can be seen, such as in the jets of rockets, in the fireball of the nuclear explosion, on the surface of space vehicles during going in or out of earth atmosphere, in the welding guns, in tube-lights and in many more systems.

1.2 Definition of Plasma

Plasma can be defined as a quasi-neutral gas which shows collective behaviour [1]. Because of quasi-neutrality condition, over a small spatial-scales and time-scales plasma has charge (i.e. ion and electron) separations but over large spatial-scales and time-scales plasma is neutral and hence there is no net electric field. This is because, the electric field of individual electron or ion is shielded by the presence of neighbouring oppositely charged particles. Hence, on a scale larger than shielding distance, the collective behaviour takes effect rather than the effect due to individual particles. This is known as Debye shielding.

1.2.1 Debye shielding and Plasma frequency

To achieve quasi-neutrality character, a small volume in plasma must have equal numbers of positively and negatively charged particles. And any localized space charge generation is followed by an immediate redistribution of opposite charged particles until the net electric field is zeroed out. This is the phenomenon of Debye shielding. And the length below which small electrostatic perturbations are present is called as the Debye shielding length (i.e. the length over which any local electrostatic perturbation is neutralized). This debye shielding length or Debye screening length is a measure of the shielding distance. [1][2]

$$\lambda_{\mathrm{D}} = \sqrt{\frac{\epsilon_0 K T_{\mathrm{e}}}{n e^2}}$$

Where, λ_{D} is the Debye screening length, ϵ_0 is the permittivity of free space, K is the Boltzmann's constant and T_{e} is the electron temperature due to its kinetic energy, and n is the particle density in plasma. If a quasi-neutral plasma is disturbed by an external perturbation, say by introduction of a single charge, the electrons being much more mobile than the ions respond to the perturbation and oscillate about the heavier and less mobile ions due to the restoring coulombic force. The frequency at which this oscillation occurs is known as the plasma frequency and is denoted by ω.

1.2.2 Plasma characterization

Broadly speaking, a plasma can be characterized by the following parameters of the plasma:

- The density of the electrons (ne) and the ions (ni). In the quasi-neutrality condition of the plasma, the density of the electrons and the ions are almost equal, i.e. ni = ne = n and n is usually called the plasma density.
- the energy distribution of the plasma particles; fe for electrons, fi for the ions and fn for the neutral particles

For an ionized gas to be termed as a plasma, it must satisfy the following criterion:

- $\lambda_{\mathrm{D}} << \mathrm{L}$
- $N_{\mathrm{D}} >>> 1$
- $\omega\tau >> 1$

Where, λ_{D} is the Debye screening length, L is the dimension of the plasma, N_{D} is the total number pf plasma particles in the Debye sphere, t is the collision time and w is the characteristic frequency of the electron oscillations

1.3 Waves and oscillations in Plasma

In a quasi-neutral plasma, whenever there is a small perturbation, an electric field builds up which in turn restores the neutrality of the plasma. This electric field restricts charge accumulation by providing a restoring force to the electrons. But because of the inertial of the electrons, they overshoot their mean position and oscillate about the position with a characteristic frequency, this mode of oscillation in plasma is called electron plasma frequency ω_{pe} which is given by [1]

$$\omega_{\mathrm{pe}} = \sqrt{\frac{e^2 n_{\mathrm{e}}}{\epsilon_0 m_{\mathrm{e}}}}$$

where, ϵ_0 and m_0 are the permittivity of free space and electron mass respectively. The ions can also oscillate at their own natural frequency called ion plasma frequency (ω_{pi}) and is given by [1]

$$\omega_{\mathrm{pi}} = \sqrt{\frac{e^2 n_{\mathrm{i}}}{\epsilon_0 m_{\mathrm{i}}}}$$

where, m_{i} is the ion mass.

In plasma with high kinetic energies i.e in warm plasmas, a low frequency wave with electron providing the restoring force and the ions contributing to the inertia is usually observed and is known as ion acoustic wave.

So, generally speaking, if the equilibrium of a plasma system is disturbed by a perturbation, then the system shows a wave like behaviour in the density gradient and hence with the potential fields. And generally, both in nature and in laboratory, the plasma system is not in an equilibrium state. And these result in creating instability in the system. These plasma instabilities can be studied by studying various oscillations in the plasma.

1.4 Plasma and Non linear dynamics

As discussed above, plasma is a highly non-linear system. Hence, the conventional tools are not sufficient to understand the nonlinear behaviour.

The linear analysis tools treat all irregular oscillations (including the chaotic ones) as stochastic processes, even though these instabilities (irregular oscillations) may have been generated by some deterministic dynamical process. Many times, we may also encounter

systems in which there is no model for analysis and we may have to study the system from the fluctuations of a single observable quantity. So, to overcome this lacking feature of the linear analysis and to analyze some complex systems we need to use nonlinear dynamics approach. However, in this report before exploring the nonlinear tools, I have thoroughly gone through the linear analysis tools. In linear analysis we generally use power spectrum analysis, auto-correlation, discrete wavelet transform, probability density function etc., and in nonlinear analysis we use Lyapunov exponent, Correlation dimension, Hurst exponent etc. Some statistical tools like normal variance, absolute mean difference, correlation coefficient etc. can also be used for characterization of coherence and stochastic resonances.

Chapter 2

The Glow Discharge Plasma Experiment

2.1 Introduction

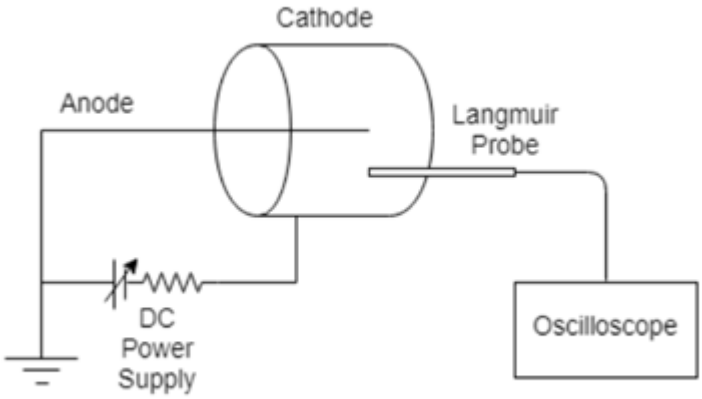
In the laboratory, plasma is generally created by supplying a large amount of energy to a neutral gas through various mechanisms, such as - electrical discharge, thermal heating, adiabatic compression, high energy beams and exothermic chemical reactions. Here in this setup, we use the electrical breakdown of the neutral gas for generating our plasma.

Due to the interaction with cosmic rays and radiations, in any neutral system there is always some amount of ionization creating electrons and ions. Hence, with the application of an electric field in a system, these electrons and ions get accelerated and collide with other neutral atoms and molecules to ionize them further. To attain a steady state, this avalanche of creation of charge particles, formed during this process is balanced by recombination and loss to container wall. In the current experiment we use a direct current (DC) discharge in neutral Argon (Ar) gas for production of plasma.

2.2 Experimental Setup: Glow Discharge Plasma

The plasma formed inside glow-discharge plasma setup are low temperature plasmas. And such plasmas are very useful for sputtering, plasma based apparatus cleaning etc. In DC glow discharge, the electrons are energized in a gas of neutral Argon atoms by the application of an electric field between the electrodes, i.e. a cylindrical cathode and an anode rod in the axis of the cylinder. The experimental setup of the electrodes is depicted in the schematic diagram shown in figure 2.1 below.

The setup consists of an outer cylindrical cathode of diameter 10 cm and of length 20 cm, and a concentric anode rod of diameter 3 mm and of length 30 cm. Electrical connections are given to the electrodes as shown in figure 2.1 above. The whole setup was mounted within a vacuum chamber which was purged out to a pressure of 0.05 mbar using a oil rotary pump. The neutral argon gas was passed through a needle valve and the pressure of the gas inside was hence controlled. The plasma in the argon gas was produced inside by application of a DC discharge voltage, which could be varied in the range 0 - 1000 Volt. The vacuum chamber which holds the electrode system is made out of stainless steel and consists of a cylindrical



setup.png

Figure 2.1: *Schematic Diagram of the cylindrical electrode system of the Glow-Discharge plasma*

main body attached with four side cylindrical bodies, the upper flange and the bottom flange. The upper and bottom flanges of diameter 30 cm and are separated by a distance of 30 cm. The four cylindrical side bodies attached to the main body are of length and diameter of 20 cm. The upper flange harbours three ports for connecting gas inlet needle valve, a Pirani gauge and a port for power connection to the electrodes. One of the side cylindrical bodies is used for mounting the electrode system and a Langmuir probe. These side cylinders also have flanges attached to them with ports allowing for insertion of the Langmuir probes. The bottom flange has a port harbouring a gate valve which connects the pipe from oil rotary vacuum pump to the vacuum chamber. The oil rotary vacuum pump has a pumping speed of 250 l/s.produced inside by application of a DC discharge voltage, which could be varied in the range 0 - 1000 Volt. The vacuum chamber which holds the electrode system is made out of stainless steel and consists of a cylindrical main body attached with four side cylindrical bodies, the upper flange and the bottom flange. The upper and bottom flanges of diameter 30 cm and are separated by a distance of 30 cm. The four cylindrical side bodies attached to the main body are of length and diameter of 20 cm. The upper flange harbours three ports for connecting gas inlet needle valve, a Pirani gauge and a port for power connection to the electrodes. One of the side cylindrical bodies is used for mounting the electrode system and a Langmuir probe. These side cylinders also have flanges attached to them with ports allowing for insertion of the Langmuir probes. The bottom flange has a port harbouring a gate valve which connects the pipe from oil rotary vacuum pump to the vacuum chamber. The oil rotary vacuum pump has a pumping speed of 250 l/s.

2.3 Verification of Paschen Curve in Glow Discharge Plasma

In this experiment, the discharge voltage applied between anode and cathode and the pressure of the gas applied are the control parameters. The Paschen curve is a plot between neutral pressure (p) and the breakdown voltage applied against anode and cathode across the gas. By varying the neutral pressure (p), the breakdown of the gas occurs at different discharge voltage as shown in the table 2.1 below and a plot of pressure (p) vs Breakdown Voltage (V_{br}) corresponding to the data table is shown in figure 2.3 below.

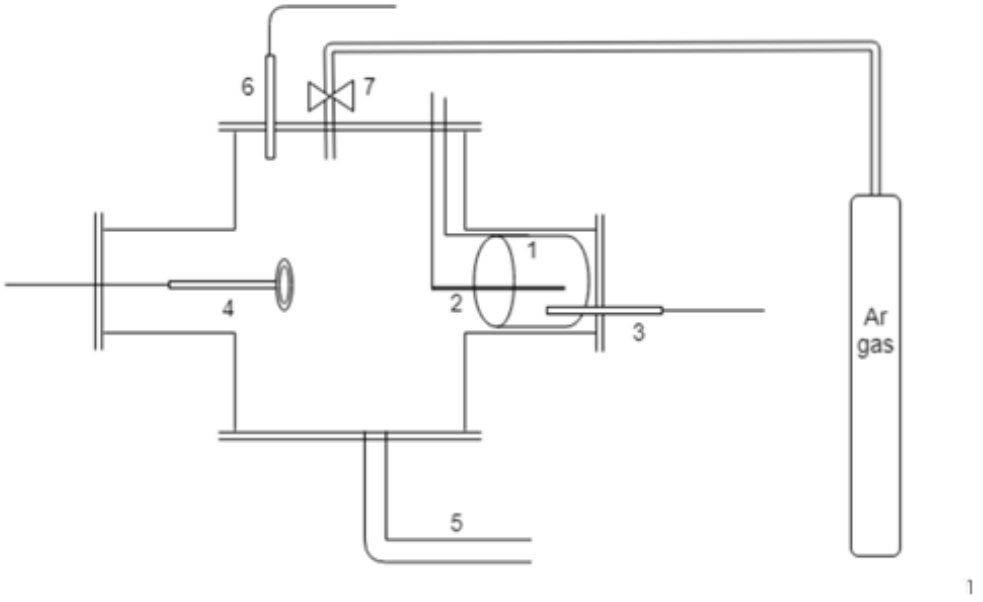


Figure 2.2: *Schematic Diagram of the Vacuum chamber setup of the Glow-Discharge plasma; 1. The cylindrical cathode, 2. The central anode rod, 3. The Langmuir probe, 4. The grid probe, 5. Connection to oil rotary vacuum pump, 6. Pirani pressure gauge probe and its connection, 7. Argon gas inlet pipe and needle valve, 8. Argon gas cylinder.*

Paschens law is an equation that relates the breakdown voltage with the applied gas pressure (p), the separation between the electrodes. But as the separation distance is constant in our experiment, here the breakdown voltage(Vbr) only depends upon the gas pressure(p). As we can see from the Paschen curve, the breakdown voltage (Vbr) initially decreases with the increase in gas pressure (p), and then begins to increase with p after going through a minimum value resembling to the typical Paschen curve.

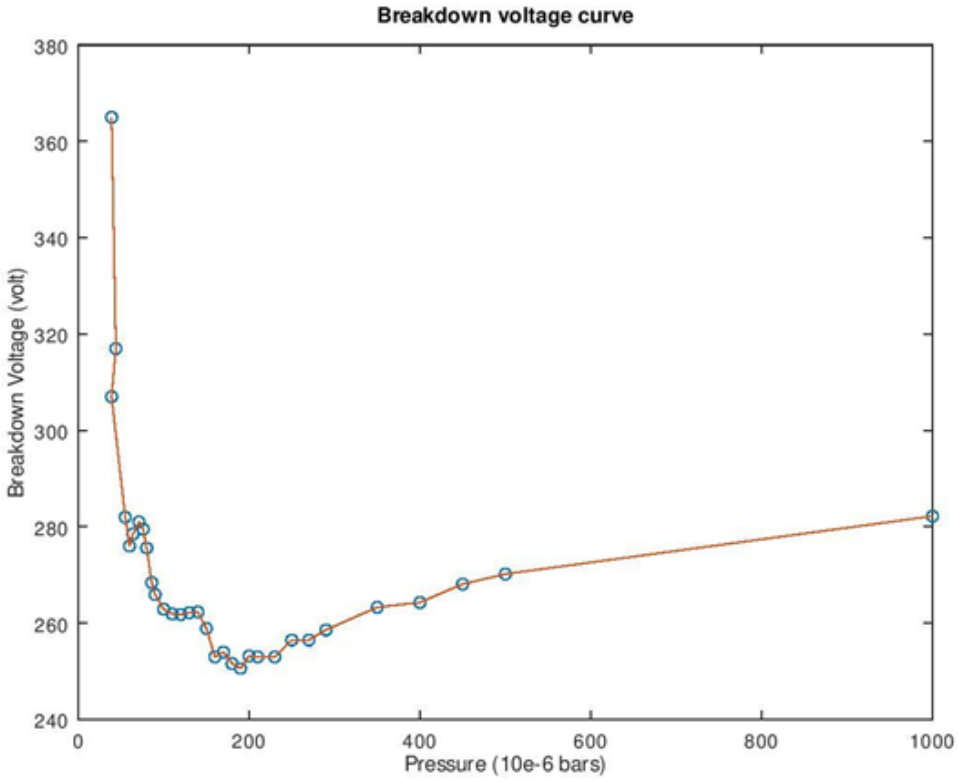


Figure 2.3: *V_{br} vs p for the experiment of verification of Paschen curve*

Pressure(p)	Discharge Voltage($V_{\rm br}$)	Pressure(p)	Discharge Voltage($V_{\rm br}$)
39	365	150	258.9
44	317	160	253
49	307	170	254
55	282	180	251.6
60	276	190	250.6
64	278.5	200	253.2
71	281	210	253
76	279.5	230	253
80	275.6	250	256.5
86	268.4	270	256.5
90	265.9	290	258.6
100	262.9	350	263.3
110	261.9	400	264.3
120	261.8	450	268.1
130	262.2	500	270.2
140	262.3	1000	282.2

Table 2.1: *Pressure (p) vs Discharge Voltage (V_{br})*

2.4 Langmuir Probe

Langmuir probe technique, first introduced by Irving Langmuir in 1924, is the most basic and widely accepted technique for measurement of plasma properties in laboratory plasma systems. A Langmuir probe is nothing but a metallic conductor introduced inside the plasma to measure various properties such as, electron density, electron temperature, plasma potential, floating potential fluctuations etc. The greatest of Langmuir probe is that it is the simplest and cheapest diagnostic that allows the local measurements of plasma parameters with high spatial and temporal resolution. However, the accuracy of Langmuir probe measurement is not so good. In-spite of this discrepancy, Langmuir probe is still used because of its simplicity and other advantages. Langmuir probe used in glow discharge plasma device (figure 2.4) is a cylindrical rod of copper with diameter 1.5 mm and length 10 mm. A Teflon coated stainless steel wire is soldered to one end of the probe and it has further has been fitted with a ceramic block. The ceramic block is then fitted with a stainless tube for support and for providing a path for a connection to the probe for actually taking measurements in terms of voltages or currents.

The information about the plasma is obtained from the Langmuir probe for measuring the current drawn by it by various applied biased voltages. Figure 2.5 shows a typical I-V characteristic of a cylindrical shaped Langmuir probe. Here V and I are the applied bias voltage to the Langmuir probe and the current drawn by the probe respectively. The I-V characteristic curve has three distinct regions: i) electron saturation region, ii) electronn retard region and iii) ion saturation region.

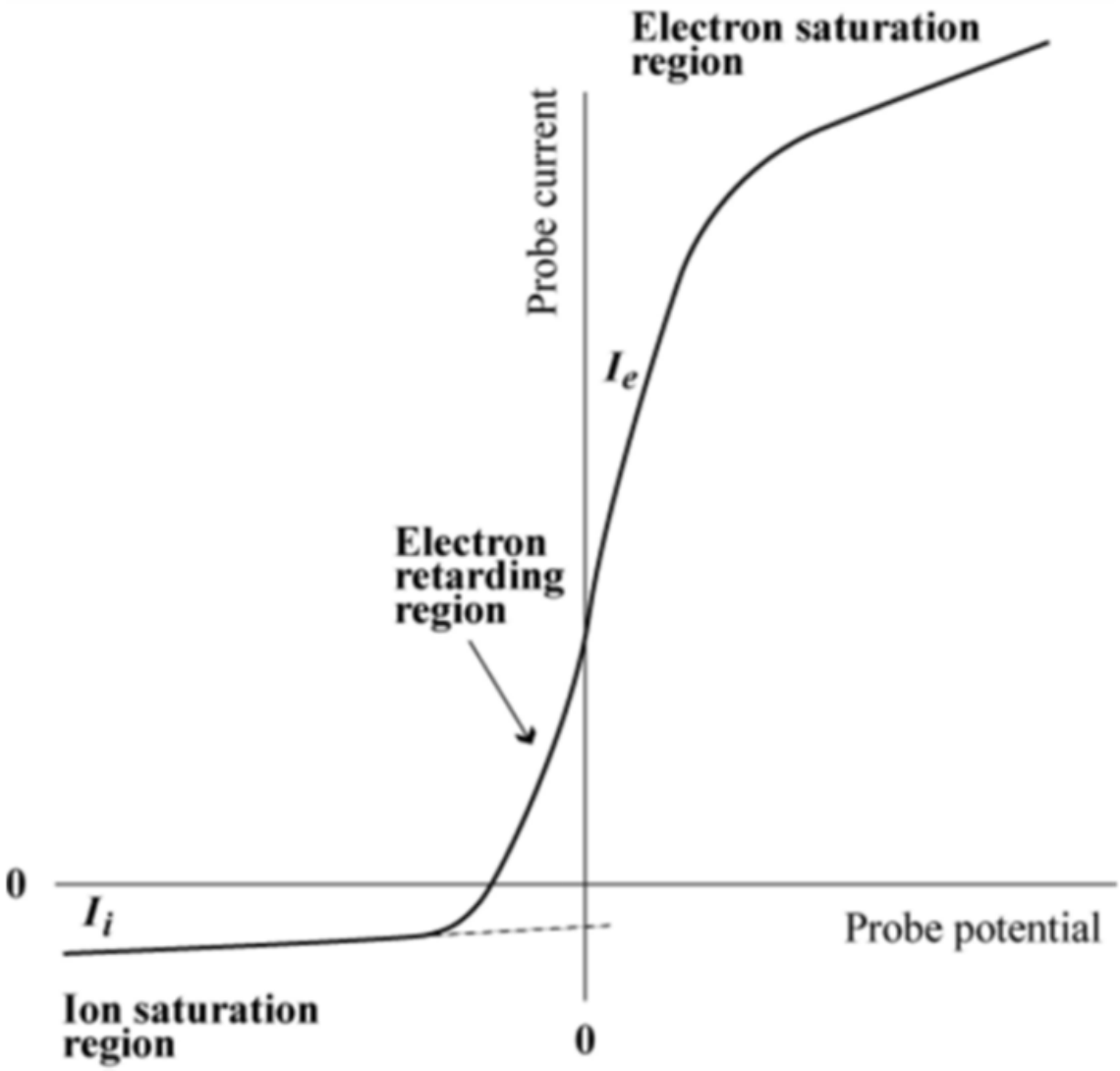


Figure 2.4: *A typical Langmuir Probes I-V characteristic curve*

2.4.1 Electron Saturation Region

When the probe bias voltage (V) is equal to the plasma potential (V_{P}) at the position of the probe, then the electric field will vanish and the perturbation to the plasma is almost non-existent. Hence, at such a situation the current drawn by the probe at this point is mainly due to the charge particles reaching the probe end solely because of their thermal velocity. And, since the thermal velocity of the electrons are much higher than that of the ions due to their lower mass, we can consider the probe current caused only by the electrons (neglecting the ion contribution). If the probe bias voltage (V) is increased above plasma potential (V_{P}), electron current does not increase further after a point since all the electrons are already being collected. This region is called electron saturation region and the

corresponding electron saturation current is given by

$$I_{\mathrm{e,sat}} = neA\sqrt{\frac{T_{\mathrm{e}}}{2\pi m_{\mathrm{e}}}}$$

where n, e, A, T_{e} and m_{e} are plasma density, electronic charge, area of Langmuir probe, electron temperature and electron mass respectively.

2.4.2 Electron Retard Region

When the probe bias voltage (V) is made negative with respect to the plasma potential V_{p}, the probe starts to repel electrons and accelerate ions resulting in the decrease of electron current and increase of ion current. Hence the total probe current decreases effectively. Finally, at sufficiently negative value of V, electron current reduces to very small fraction of I_{e},sat and overall current becomes equal to the ion current. At this potential total current drawn by the probe is zero and the potential is called the floating potential V_{f}. An insulated probe placed inside the plasma would assume this floating potential. This is because of that an insulated probe inside the plasma is

$$I_{\mathrm{e}} = I_{\mathrm{e,sat}} e^{\frac{e(V-V_{\mathrm{p}})}{T_{\mathrm{e}}}}$$

2.4.3 Ion Saturation Region

When probe-bias voltage (V) is negative enough to repel all the electrons then the total current collected by the probe is ion current. Probe current remains constant with the further decrease of V resulting in a saturation region. This region is called ion saturation region. the corresponding Ion saturation current is obtained from the Bohm sheath criterion and is given by,

$$T_{\mathrm{e,sat}} = 0.61neA\sqrt{\frac{T_{\mathrm{e}}}{m_{\mathrm{i}}}}$$

where m_{i} denotes ion mass and all other symbols have their usual meanings.

Chapter 3

Data Analysis Techniques: Time Series Analysis

3.1 Introduction

A time series data is a collection of observations taken sequentially and in equally spaced periods in time. These observations are crucial for the studies of the dynamic behavior of the system of interest. These time series data contain the information about some measured physical quantity over equally spaced time. By careful analysis of these time series, useful information about the dynamic behavior of the system can be obtained. It is important to note that, unlike the statistical ensemble approach (where each observation is independent of one another), these successive observations in a time series data set are not independent observations. And hence, these time-series data are essential for the study of underlying dynamics of a system.

In order to explore plasma dynamics, plasma fluctuations data have been collected and analyzed. Since these fluctuations are mainly generated due to various linear and non-linear interaction between plasma components, these generally are oscillatory in nature. The origin of these oscillations may be mainly due to: 1) interplay of internal complex system parameters, called self-excited oscillations and 2) external forcing, called forced oscillations. In this report, we discuss the analysis of self-excited oscillations of plasma. In the both the experiments, we have made sequential observation of floating potential and recorded respective time series. These time series are analyzed using various linear and non-linear time series analysis tools along with some statistical tools in both time as well as frequency domain. The methodology of various time-series analysis tools used are given in this chapter.

3.2 Power Spectral Method

The Fourier theorem makes the basis of power spectrum analysis method.

3.2.1 The Fourier Theorem

Any real-world physical process can be expressed as a function of time (time domain) or as a function of frequencies (frequency domain). Fouriers theorem states that any such waveform

can be represented as a weighted sum of sine waves and cosine waves. These sinusoids, in general, are referred to as harmonics of the waveform, except one sinusoidal function among them that has the same frequency as the original waveform is called the fundamental.

3.2.2 The Fourier Transform

A physical process expressed in time domain can be deconstructed into a frequency domain with the help of Fourier Transform and vice-versa with Inverse Fourier transform. The Fourier transform equations can be represented as:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t} dt$$

where f $\epsilon(-\infty, \infty)$

And frequency domain transform can be represented as:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{2\pi i f t} df$$

where t $\epsilon(-\infty, \infty)$

3.2.3 Nyquist Critical Frequency

For any sampling interval Δt there exists a corresponding frequency f_{c}, called the Nyquist critical frequency, and is given by

$$f_{\mathrm{c}} = \frac{1}{2\Delta t}$$

This suggests that the sampled data x(t) with an time interval Δt (i.e sampling rate $1/\Delta t$) is bandwidth limited to frequencies smaller than the Nyquist critical frequency f_{c}. So, the transformed X(f) is obtained at values of f_{k} = -f_{c} to f_{c}.

$$f_{\mathrm{k}} = \frac{k}{N\Delta t}$$

where, k ranges from -N/2 to N/2.

3.2.4 Fast Fourier Transform

The Fast Fourier Transform (FFT) is an optimized way to do Discrete Fourier Transform (DFT) that takes less computation time. For N data samples the DFT using just the DFT equation is computed in N^2 operations; whereas with FFT algorithm (due to Cooley and Tukey) the same task can be performed in $Nlog_2N$ operations.

From the Linearity property of Discrete Fourier Transform, it can be showed that a Discrete Fourier transform of length N can be re-written as a sum of two Discrete Fourier Transform of length N=2 (Danielson and Lanczos). First part is the sum of odd numbered points and second is sum of even numbered points.

$$X_{\mathrm{k}} = \sum_{n=0}^{N-1} x_{\mathrm{n}} e^{-2\pi i n k/N}$$

$$X_{\mathrm{k}} = \sum_{n=0}^{N/2-1} x_{2\mathrm{n}+1} e^{-2\pi i (2n+1) k/N} + \sum_{n=0}^{N-1} x_{2\mathrm{n}} e^{-2\pi i (2n) k/N}$$

$$X_{\mathrm{k}} = e^{-2\pi i k/N} \sum_{n=0}^{N/2-1} x_{2\mathrm{n}+1} e^{-2\pi i (2n+1) k/(N/2)} + \sum_{n=0}^{N-1} x_{2\mathrm{n}} e^{-2\pi i (2n) k/(N/2)}$$

This can then be performed recursively to obtain the FFT.

In this algorithm, there is a restriction that the original numbers of samples N must be an integer power of 2. When the numbers of samples are not a power of two, the data is padded with zeros till next power of 2. A periodic or quasi-periodic signal shows sharp spectral line in power spectrum plot, whereas for a chaotic signal, it will show a broad band. This can be shown in the following examples.

Case-i: Sine wave

Consider the data sampled at every 2 Microseconds; i.e. with a sampling rate of 500000 Hz. And let the sample size be of 500000 so that the signal spans for about a second. For Test Case-1 Lets consider a simple sinusoidal wave of 1000 Hz.

$$X = sin(2\pi 1000T)$$

Such a data will show a simple sinusoidal wave. A graph can be plotted to show a part of this signal as in figure 4.1 (a). As the condition of FFT the original data should be integer power of 2, the original data had to be padded with 0 from 500001 to 524288 (i.e. next integer power of 2) while extending the time axis. After obtaining Fourier transform of this signal the Frequency vs. Magnitude is plotted. The maximum frequency that can be resolved with this data is 250000 Hz (equal to Nyquist critical frequency). For better viewing the data, here frequency of only up to 5000 Hz is plotted in figure 4.1 (b).

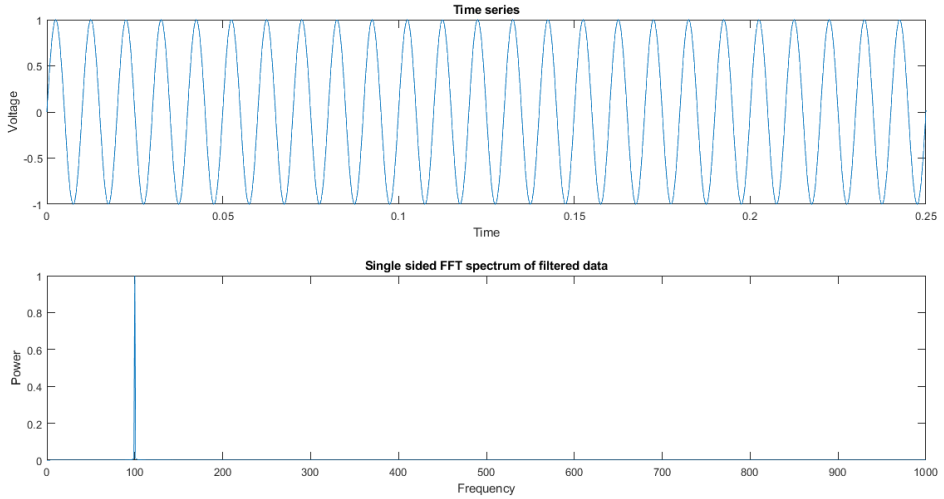


Figure 3.1: *(a) time domain and (b) frequency domain plot of a sample sine wave of frequency 1000 Hz*

Case-ii: Sine wave superposition

For Test Case-2 consider the following signal generating function which is a sum of two sin waves, one with frequency 1000 Hz and another with frequency 2000 Hz.

$$X = sin(2\pi 1000T) + sin(2\pi 2000T)$$

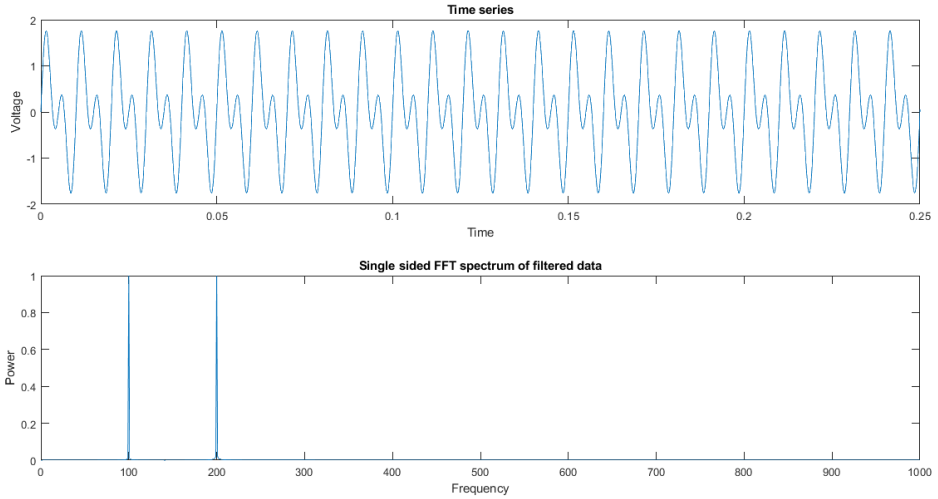


Figure 3.2: *(a) shows sine wave superposed with two sine waves of frequency 1000Hz and 2000Hz respectively (b) After obtaining FFT we can see the frequencies of the mixture*

Case-iii: Saw Tooth signal

A Saw tooth waveform consists of integer numbered harmonics in addition to the fundamental with amplitudes inversely proportional to their harmonic integer number, and can be obtained by a sum of sine waves as follows:

$$X = \sum_{k=0}^{\infty} \frac{1}{k} sin(2\pi k 500T)$$

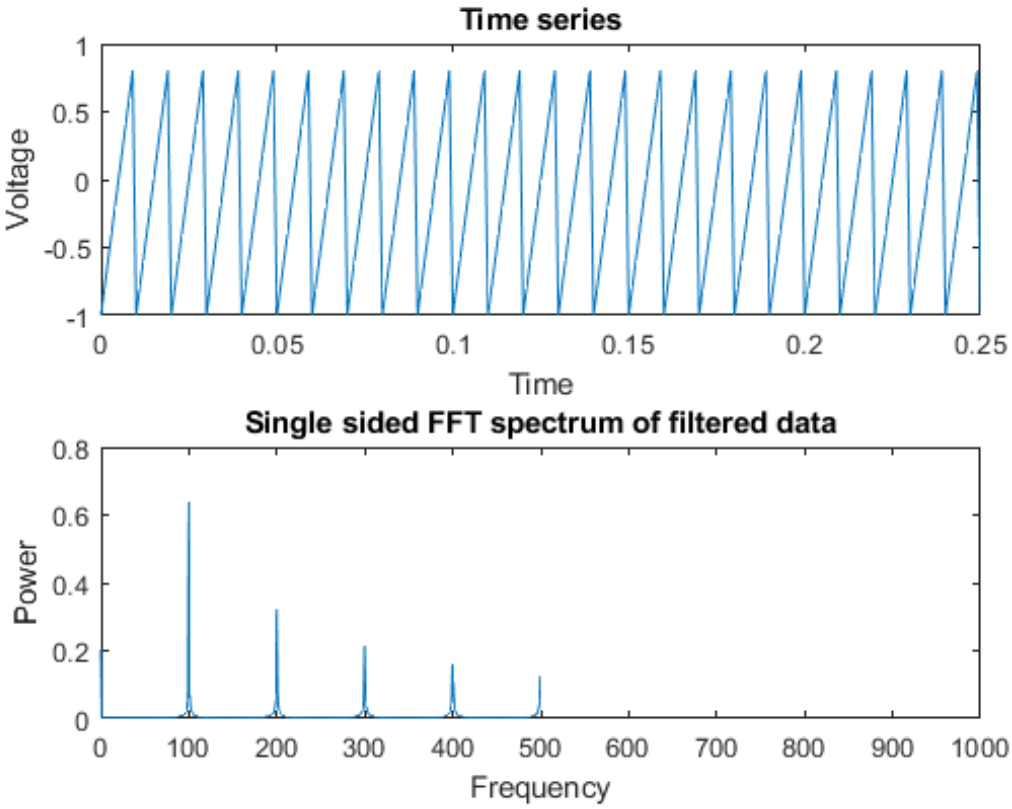


Figure 3.3: *((a) shows the sawtooth wave signal (c) After obtaining FFT we can see the frequencies of the mixture*

Case-iv: Non-linear signal

Here, lets consider a non-linear ion-acoustic wave in plasma. Unlike the previous cases where all the frequencies were integral multiples of the fundamental frequency, here the frequencies are scattered in the frequency domain.

The power spectrum method will be extensively used to determine the characteristic modes and the dominant frequencies present in the plasma fluctuations. The periodic or quasi-periodic signal shows sharp peaks in the spectral lines, where as for a chaotic it will show a broad-band.

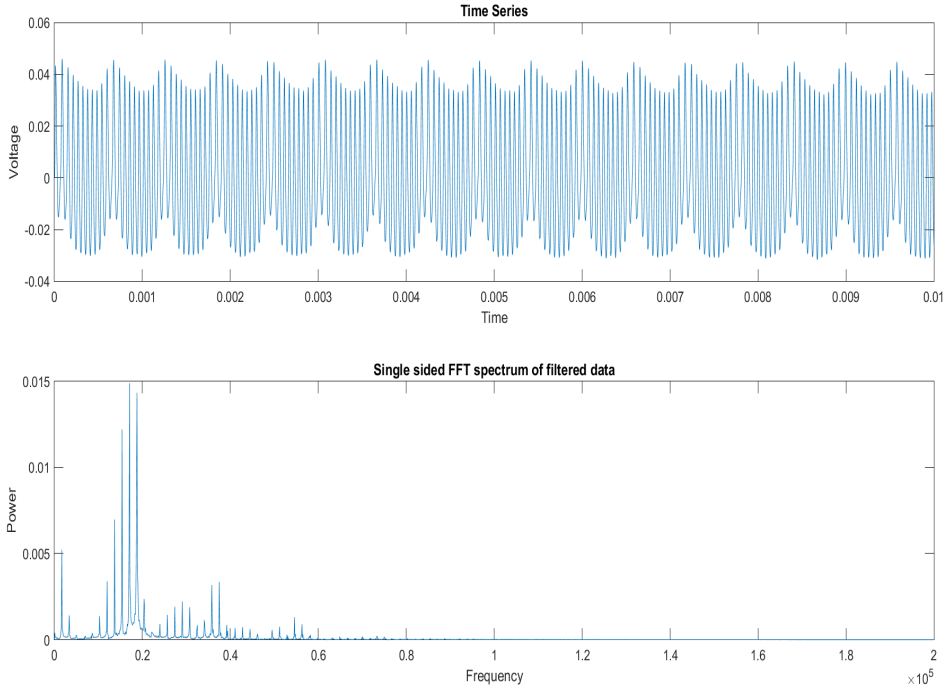


Figure 3.4: *((a) shows the time series data in time domain and (b) shows the FFT of the same time series*

3.3 Phase Space Reconstruction

Real world process evolution are usually observed in a scalar time-series with a single observable, which generally does not provide all the possible states of the system and related phase space related information. Hence, in order to distinguish whether a signal belongs to periodic, irregular or chaotic system, to observe nonlinear structures and to estimate correlation dimension, Lyapunov exponent etc., the time-series is needed to be described in a suitable phase space. This is an analysis tool which describes the time-series in a geometric form.

From the single variable scalar time series measurements, the phase space vectors can be reconstructed using the method of delay. According to Takens theorem, from a given time-series data, $x_0, x_1, x_2, ..., x_{\mathrm{N}}$, where N is the total number of observations in time $\mathrm{N}\Delta t$ with Δt being the time spacing of measurement. In the above time series data xi denotes the observation at time $\mathrm{i}\Delta t$. The reconstructed phase space vector for such a time-series can be constructed by time delay embedding as:

$$X_{\mathrm{i}} = [x_{\mathrm{i}}, x_{\mathrm{i+T}}, x_{\mathrm{i+2T}}, ..., x_{\mathrm{i+(m\text{-}1)T}}]$$

where, T is the embedding time delay between two consecutive component of the delay vector X_{i}. This embedding time delay can be estimated using the mutual information method as suggested by Fraser and Swinney. And, m is the embedding dimension, the minimum embedding dimension can be estimated by the false nearest neighbour method proposed by Kennel et al.

Using this set of phase space vectors (X_i), one can draw a trajectory in the phase space. This trajectory is useful to estimate the correlation dimension and Lyapunov exponents of the time series data.

Phase space trajectories of various types of signals

Case-1: Periodic signal

For periodic signal where the value of the variable shows simple sinusoidal repetition, the shape of the phase portrait diagram is a simple elliptical orbit.

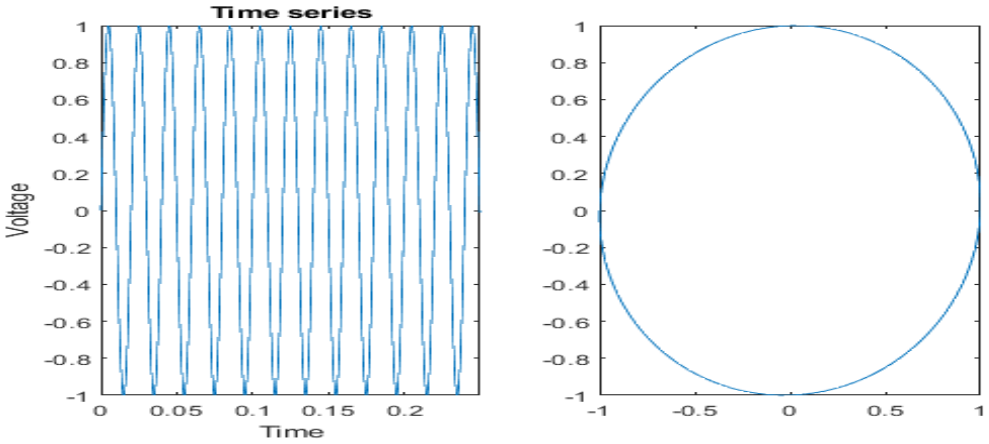


Figure 3.5: *(a) A Periodic sinusoidal signal (b) Phase portrait of the periodic signal*

Case-2: Quasi-periodic signal

For Quasi-periodic signals where the value of the variable shows complex repetitions with time, the phase portrait diagram has a shape of a complex orbit.

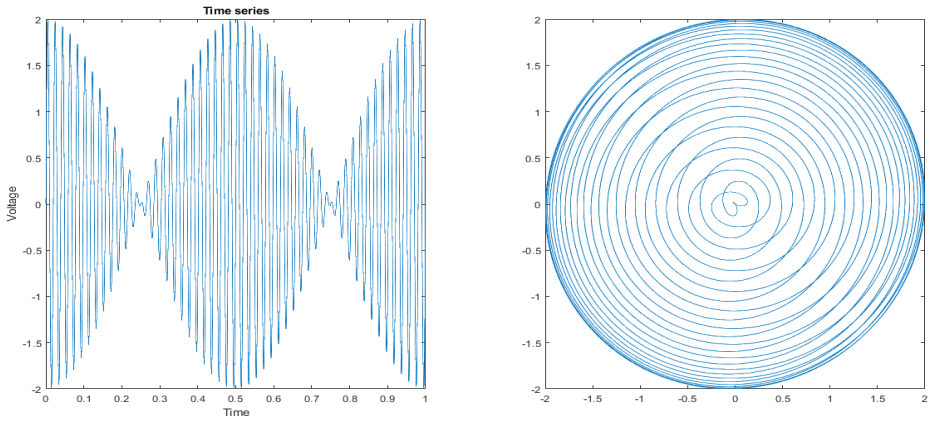


Figure 3.6: *(a) A Quasi-periodic signal (b) Phase portrait of the quasi-periodic signal*

Case-3: Periodic signal mixed with random noise

When a periodic signal is mixed with random noise the phase portrait takes a shape of an ellipse with a noisy outline suggesting presence of a random noise.

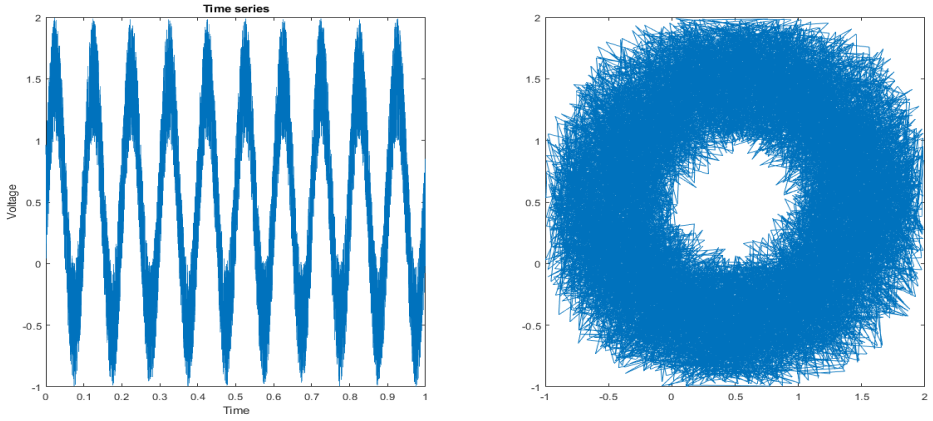


Figure 3.7: *(a) A noisy periodic signal (b) Phase portrait of the noisy periodic signal*

Case-4: Non-linear Signal

For Non-linear (or chaotic) signals the phase space has specific strange looking shapes called Attractors.

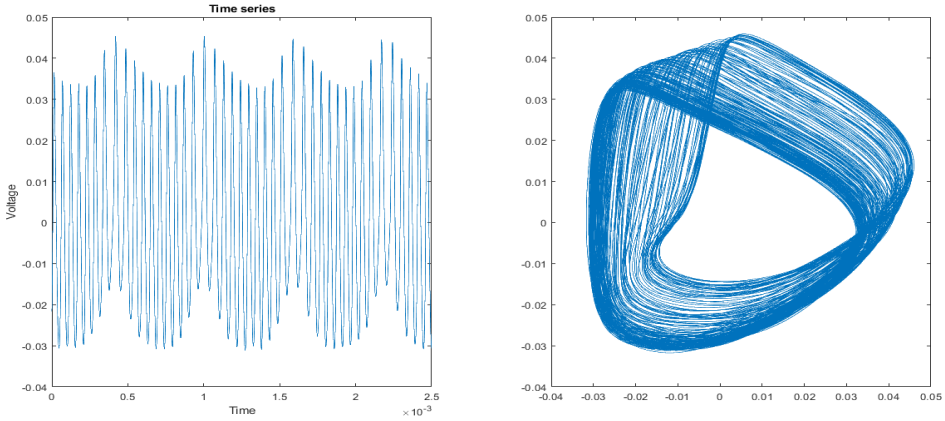


Figure 3.8: *(a) A non-linear signal (b) Phase portrait of the non-linear signal*

The autocorrelation function also provides a diagnostic tool for discriminating between periodic and stochastic behavior. In a periodic signal the autocorrelation is periodic, whereas in a stochastic signal the autocorrelation will be irregular. The autocorrelation function is defined by

$$c_{\mathrm{j}} = \frac{1}{N}\sum_{i=1}^{N} y_{\mathrm{i}} y_{\mathrm{i+j}}$$

where periodic boundary conditions

$$y_{\mathrm{N+k}} = y_{\mathrm{k}}$$

are imposed to extend the times series beyond y_{N}. This function provides a simple quantitative measure of the linear correlation between data points.

3.3.1 Optimal Embedding Dimension

The minimum embedding dimension in this project was determined by a method suggested by Cao *et al.* The method is described below.

Suppose that we have a time series $X_1, x_2, .., X_{\mathrm{N}}$. The time-delay vectors can be reconstructed as follows:

$$y_{\mathrm{i}}(d) = (x_{\mathrm{i}}, x_{\mathrm{i+T}}, ..., x_{\mathrm{i+(d\text{-}1)T}})$$

$$i = 1, 2, ..., N - (d-1)T$$

....(1)

where d is the embedding dimension and r is the time delay. Note that $y_{\rm i}(d)$ means the $i_{\rm th}$ reconstructed vector with embedding dimension d. Similar to the idea of the false neighbor method, we define

$$a(i,d) = \frac{||y_{\rm i}(d+1) - y_{\rm n(i,d)}(d+1)||}{||y_{\rm i}(d) - y_{\rm n(i,d)}(d)||}$$

$$i = 1, 2, ..., N - dT$$

where ||.|| is some measurement of Euclidian distance and is given in this paper by the maximum norm, i.e.,

$$||y_{\rm k}(m) - y_{\rm l}(m)|| = max|x_{\rm k+jT} - x_{\rm l+jT}$$

$y_{\rm i}(d+1)$ is the $i_{\rm th}$ reconstructed vector with embedding dimension d + 1. i.e.. $y_{\rm i}(d+1) = (x_{\rm i}, x_{\rm i+T}, ..., x_{\rm i+dT})$; n(i, d) (1 ¡ n(i, d) ¡ N-dT) is an integer such that $y_{\rm n(i,d)}(d)$ is the nearest neighbor of $y_{\rm i}(d)$ in the d-dimensional reconstructed phase space in the sense of distance ||.|| we defined above. n(i, d) depends on i and d.

If d is qualified as an embedding dimension by the embedding theorems [1,2], then any two points which stay close in the d-dimensional reconstructed space will be still close in the (d + 1)-dimensional reconstructed space. Such a pair of points are called true neighbors, otherwise, they are called false neighbors. Perfect embedding means that no false neighbors exist. This is the idea of the false neighbor method in [7], where the authors diagnosed a false neighbor by seeing whether the a(i, d) is larger than some given threshold value. The problem is how to choose this threshold value. From the definition of a(i, d) in [7] or Eq. (1), one can see that the threshold value should be determined by the derivative of the underlying signal, therefore, for different phase points i, a(i, d) should have different threshold values at least in principle. Furthermore, different time series data may have different threshold values. These imply that it is very difficult and even impossible to give an appropriate and reasonable threshold value which is independent of the dimension d and each trajectory's point, as well as the considered time series data. To avoid the above problem, we instead define the following quantity, i.e., the mean value of all a (i, d)'s,

$$E(d) = \frac{1}{N - dT} \sum_{i=1}^{N-dT} a(i,d)$$

E (d) is dependent only on the dimenston d and the lag r. To investigate its variation from d to d 4- 1. we define

$$E1(d) = \frac{E1(d+1)}{E1(d)}$$

E1(d) stops changing when d is greater than some value $d_{\rm o}$ if the time series comes from an attractor. Then $d_{\rm o}$+1 is the minimum embedding dimension we look for.

This method was used to find the embedding dimension of the time series data collected from the non-magnetic plasma. Two such time series data are given below with their embedding dimensions.

Data 1: The time series and its FFT are given as follows:

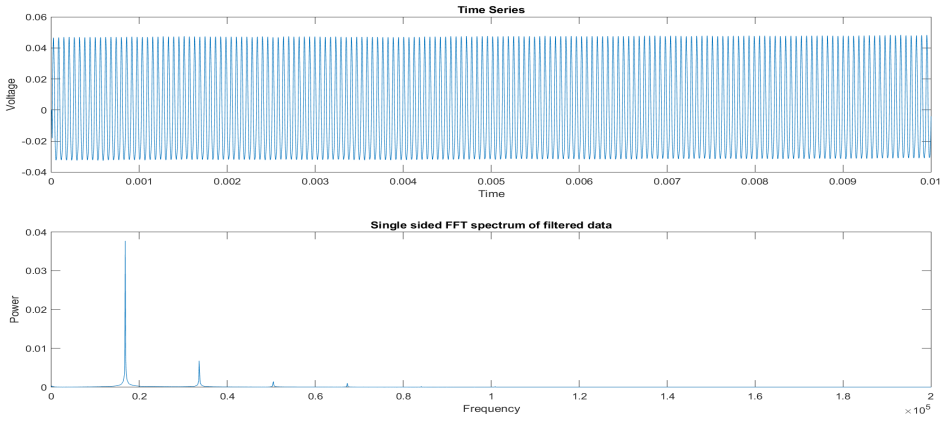


Figure 3.9: *(a) A non-linear signal from the plasma system (b) Phase portrait of the non-linear signal*

The variation of E1 and E2 with respect to embedding dimension of the given time series data is given as follows:

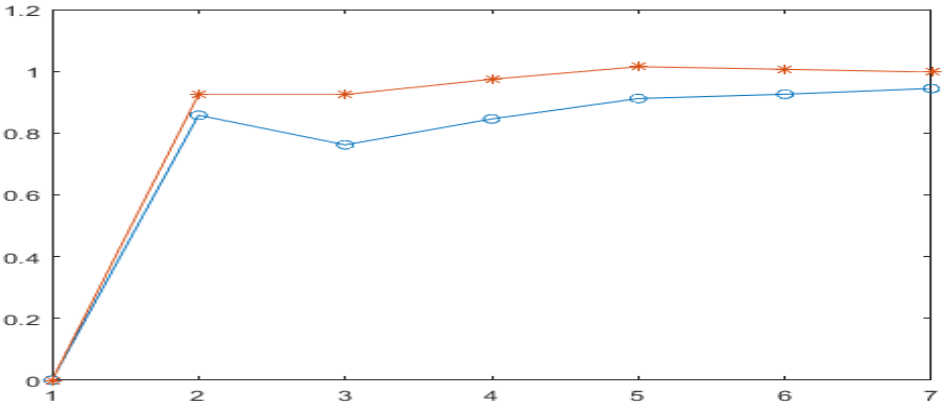


Figure 3.10: *Variation of E1 and E2 with respect to embedding dimension of the given time series data above, x-axis is (d) and y-axis is E1 E2*

Here we can see that E1 and E2 do not change much after d = 2, hence from Cao's proposed method the optimum embedding dimension for here is 2+1=3.

Data 2: The time series and its FFT are given as follows:

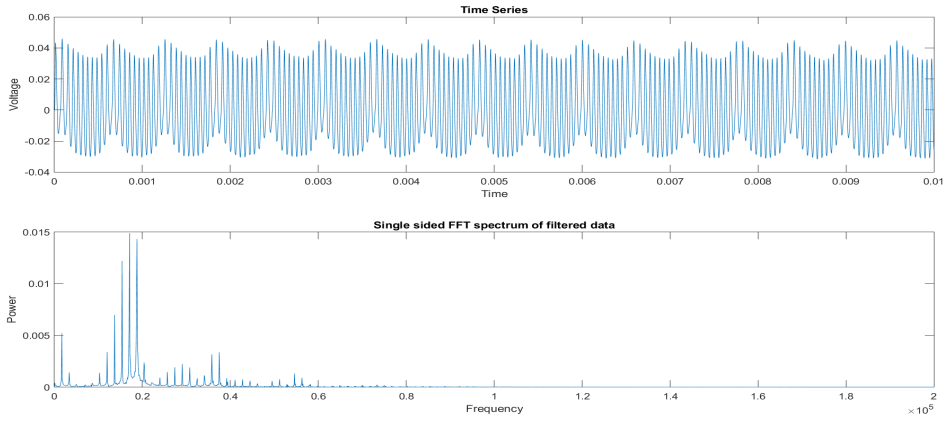


Figure 3.11: *(a) A non-linear signal from the plasma system (b) Phase portrait of the non-linear signal*

The variation of E1 and E2 with respect to embedding dimension of the given time series data is given as follows:

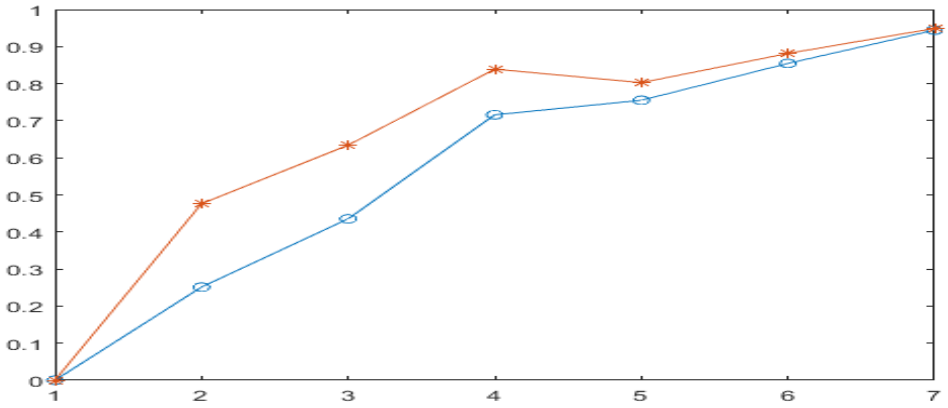


Figure 3.12: *Variation of E1 and E2 with respect to embedding dimension of the given time series data above, x-axis is (d) and y-axis is E1 E2*

Here d = 4, hence from Cao's proposed method the optimum embedding dimension for here is 4+1=5.

3.4 Recurrence Plots

In descriptive statistics and chaos theory, a recurrence plot (RP) is a plot showing, for each moment i in time, the times at which a phase space trajectory visits roughly the same area in the phase space as at time i. In other words, it is a graph of

$$\vec{x}(i) \approx \vec{x}(j)$$

showing i on a horizontal axis and j on a vertical axis, where x $\vec{x}$ is a phase space trajectory.

Eckmann et al. (1987) introduced recurrence plots, which provide a way to visualize the periodic nature of a trajectory through a phase space. Often, the phase space does not have a low enough dimension (two or three) to be pictured, since higher-dimensional phase spaces can only be visualized by projection into the two or three-dimensional sub-spaces. However, making a recurrence plot enables us to investigate certain aspects of the m-dimensional phase space trajectory through a two-dimensional representation.

A recurrence is a time the trajectory returns to a location it has visited before. The recurrence plot depicts the collection of pairs of times at which the trajectory is at the same place, i.e. the set of {(i , j) (i, j) with $\vec{x}(i) = \vec{x}(j)$. This can show many things: for instance, if the trajectory is strictly periodic with period T , then all such pairs of times will be separated by a multiple of T and visible as diagonal lines. To make the plot, continuous time and continuous phase space are discretized, taking e.g.$\vec{x}(i)$ as the location of the trajectory at time (iτ and counting as a recurrence any time the trajectory gets sufficiently close (say, within) to a point it has been previously. Concretely then, recurrence/non-recurrence can be recorded by the binary function

$$R(t, \tau) = H(\epsilon - ||f(t) - f(\tau)||$$

where H(x) is the Heaviside step function and $||f||$ denotes norm

Caused by characteristic behaviour of the phase space trajectory, a recurrence plot contains typical small-scale structures, as single dots, diagonal lines and vertical/horizontal lines (or a mixture of the latter, which combines to extended clusters). The large-scale structure, also called texture, can be visually characterised by homogenous, periodic, drift or disrupted. The visual appearance of an RP gives hints about the dynamics of the system.

The recurrence plots for the data collected from the plasma are given below.

Data 1: The following recurrence plot is for the time series data shown in **Figure 3.9**

Figure 3.13: *Recurrence plot of the time series data non-linear signal in Figure 3.9*

Data 2: The following recurrence plot is for the time series data shown in **Figure 3.11**

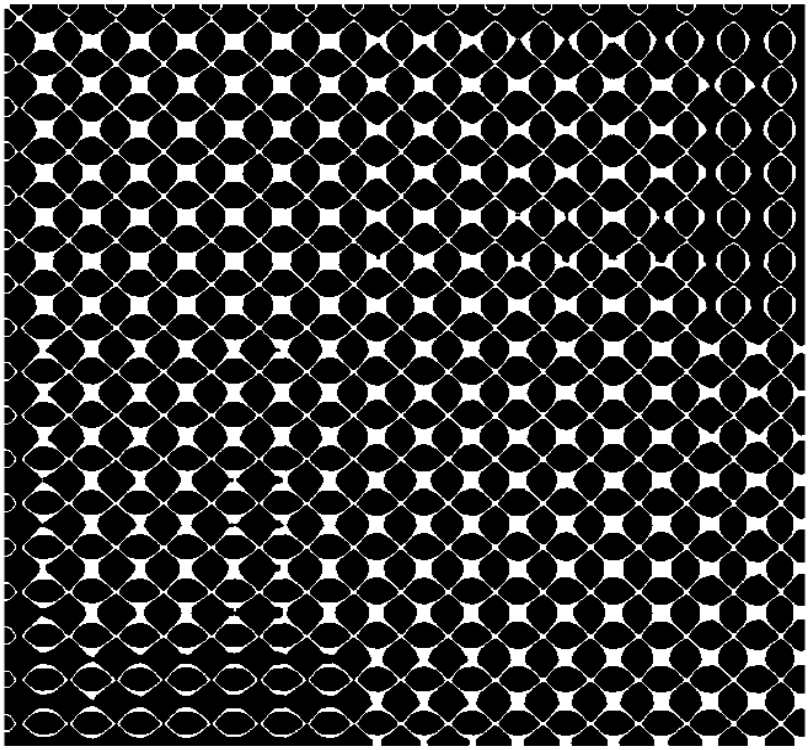


Figure 3.14: *Recurrence plot of the time series data non-linear signal in Figure 3.11*

3.5 Filtering

Filtering of a time series data is needed in order to eliminate or minimize the noise in the signal dynamical signal may be captured accurately. In this project, Singular Value Decomposition (SVD) has been used for filter ting the time series data.

3.5.1 Singular Value Decomposition (SVD)

A useful method of reconstruction of phase portraits based on SVD was introduced by Broomhead *et al.* This method provides phase portrait reconstruction with little noise reduction From the measured discrete time series $(V_{\mathrm{i}}|i, ..., N_{\mathrm{T}})$ where N_{T} is the number of data points a sequence of vectors $(x_{\mathrm{i}}|i, ..., N)$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1^T \\ \mathbf{X}_2^T \\ \vdots \\ \mathbf{X}_N^T \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_2 & v_3 & \cdots & v_{3n+1} \\ \vdots & & \ddots & \vdots \\ v_N & v_{N+1} & \cdots & v_{N+n-1} \end{bmatrix} \qquad (3)$$

where $N = N_{\mathrm{t}} - (n-1)$ Note that the matrix X is the pseudo-phase portrait by the method of delays in n dimensional pseudo-phase space with a delay time of one unit The SVD of the trajectory matrix gives

$$X = S\Sigma C^{\mathrm{T}} ....(4)$$

where S is the Nxn matrix of eigenvectors of XX^{T} and $N \gg n$ C is the nxn matrix of eigenvectors of $X^{\mathrm{T}}X$ and S is the nxn diagonal matrix consisting of singular values $\mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_{\mathrm{n}})$, Rearranging equation (4)

$$XC = S\Sigma ....(5)$$

The matrix XC is the trajectory matrix projected onto basis (c_{i}) where c_{i} is the i_{th} column of C. One can think of the trajectory as exploring, on average, an n-dimensional ellipsoid, where c_{i} represent directions and σ_{i} represent the lengths of the principal axes of the ellipsoid. The main concept of this method is to extract the dimensionality n' (minimum embedding dimension) of the subspace containing the embedded manifold, where, $n' \leq n$. The dimensionality n' is the rank of the eigenvector matrices [rank(S)=rank"C], where the rank is the number of non-zero singular values. At this point, one can intuitively think of the physical meaning of the dimensionality n' as an effective embedding dimension. In other words, the matrix XC with embedding dimension n' has no less information than the matrix with embedding dimension n. Also, note that the SVD ensures that each column of the

matrix XC is linearly independent. In the presence of noise, the noise causes all the singular values of the trajectory matrix to be non!zero[However, assuming the noise is white, the noise will cause all the singular values of X to be shifted uniformly, i.e., they can be written as

$$\sigma_i{}^2 = \bar{\sigma}_i{}^2 + \sigma_{\text{noise}}{}^2$$
$$i = 1, 2, ..., k$$
$$\sigma_k{}^2 =, ..., = \sigma_n{}^2 = \sigma_{\text{noise}}{}^2 .....(5)$$

where σ_{noise} are the singular values of the noise floor, and the trajectory matrix can also be written as

$$\mathbf{X} = \bar{\mathbf{X}} + \mathbf{N} = [\mathbf{S}_1 \quad \mathbf{S}_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{C}_1^T \\ \mathbf{C}_2^T \end{bmatrix} \qquad (7)$$

where $\bar{\sigma}$ is the deterministic part of the trajectory matrix, N is the noise!dominated part, In order to separate the noise!dominated part from the trajectory matrix, one can estimate the deterministic part $\bar{\sigma}$ by either least squares or minimum variance estimate. The least squares estimate of $\bar{\sigma}$ is given by

$$\bar{X}_e = S_1 \Sigma_1 {C_1}^T ....(8)$$

The SVD analysis and the reconstructed time series is given below

Original Time series:

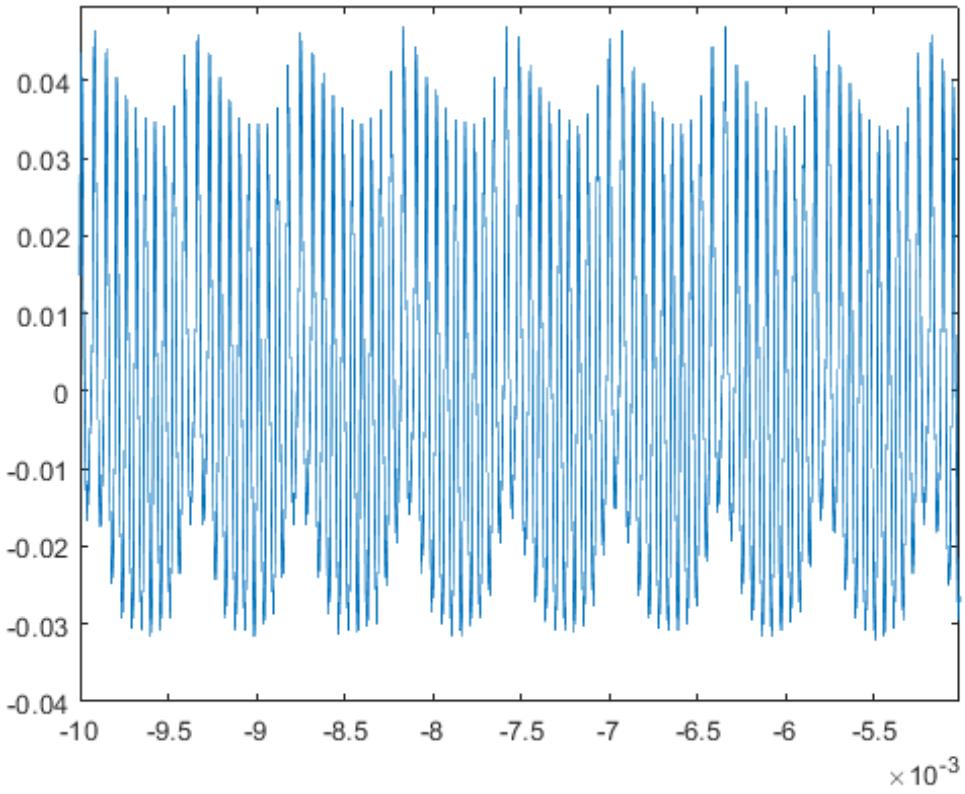


Figure 3.15: *Original Time Series of the signal from the plasma system*

Original FFT:

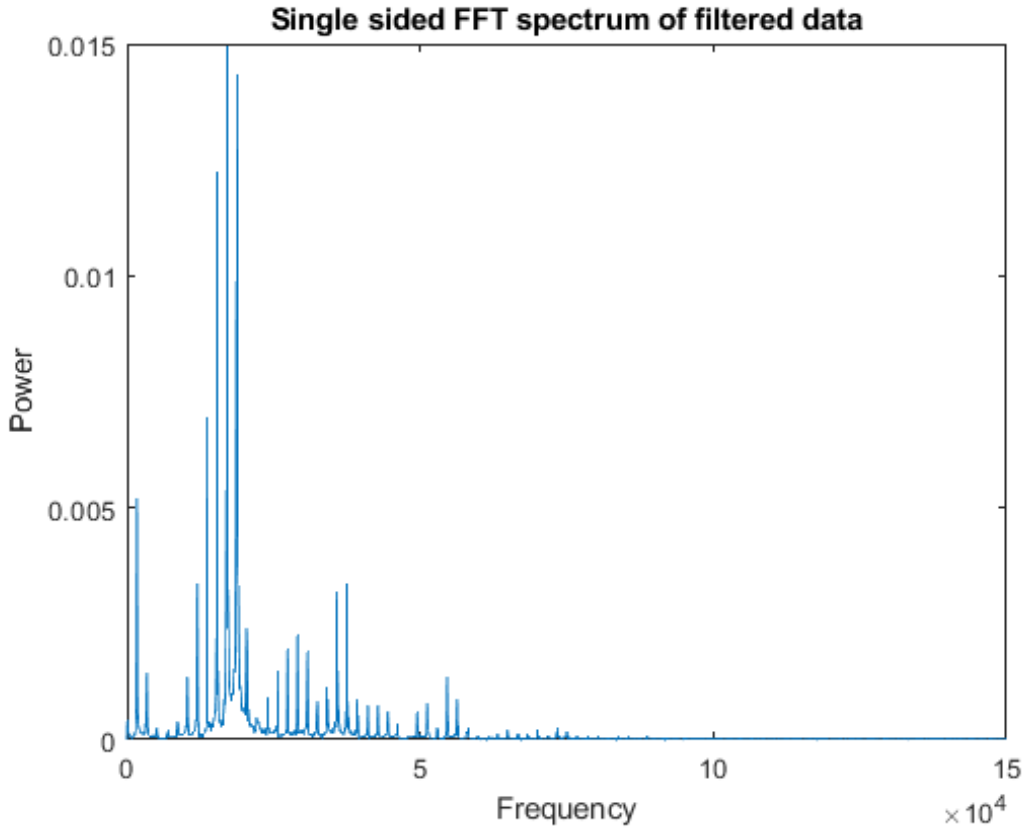


Figure 3.16: *FFT of the unfiltered signal from the plasma system*

log-log plot of original FFT:

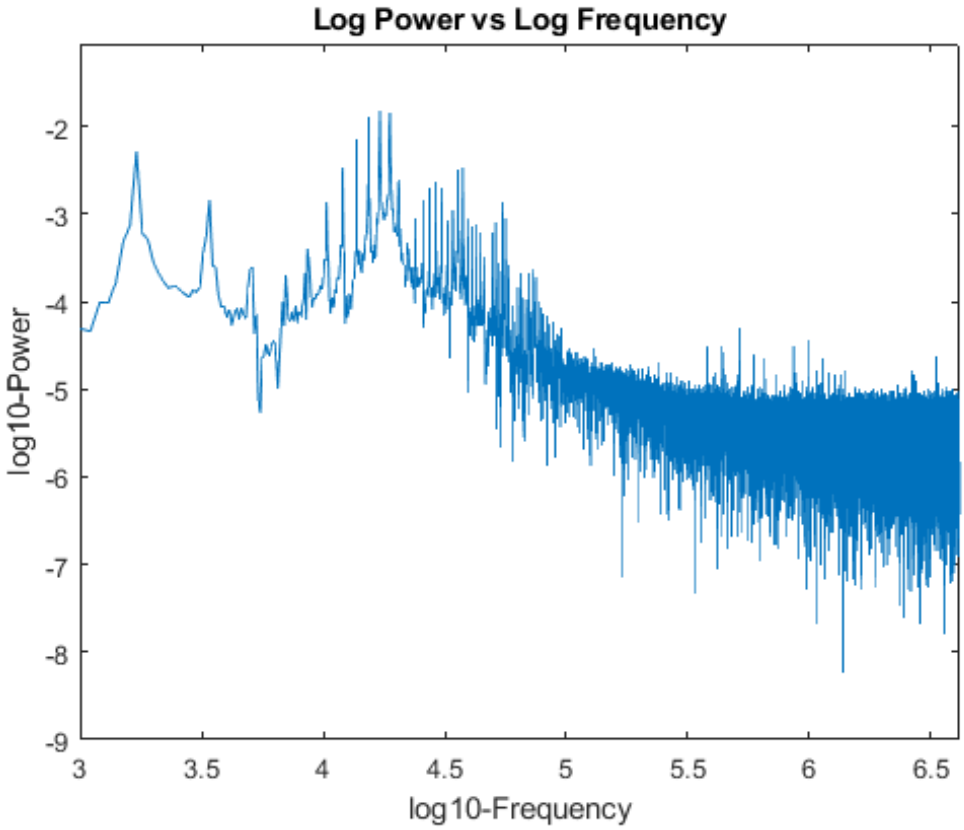


Figure 3.17: *log-log FFT of the unfiltered signal from the plasma system*

Phase portrait of original FFT:

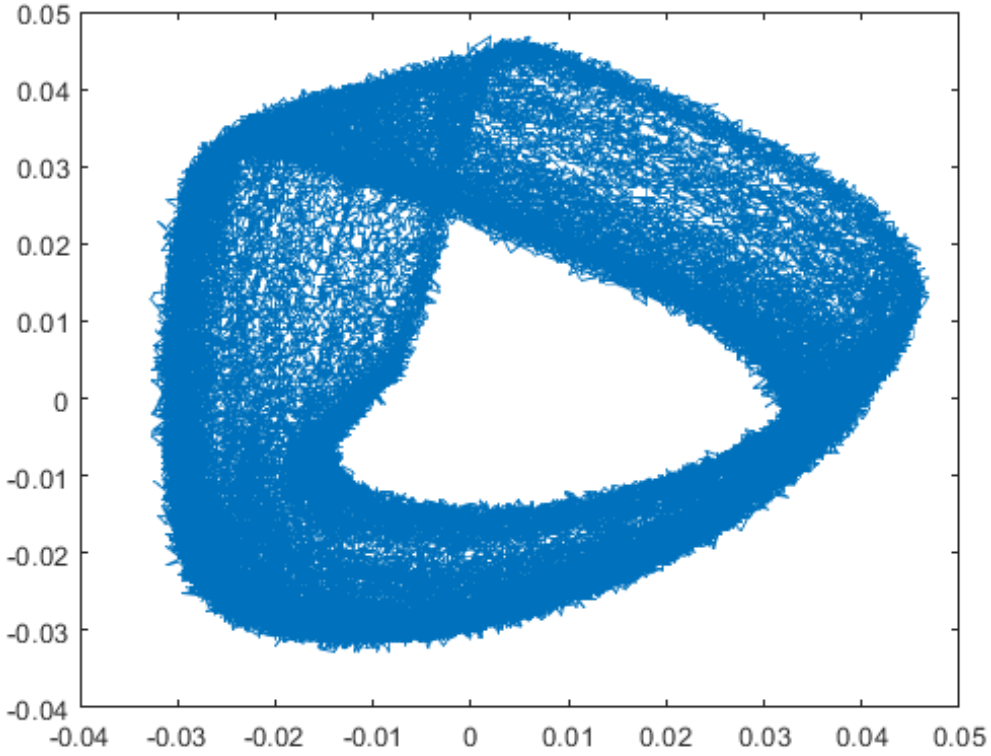


Figure 3.18: *Phase portrait of the unfiltered signal from the plasma system*

SVD analysis for embedding dimension (d)=4:

FFT after SVD analysis:

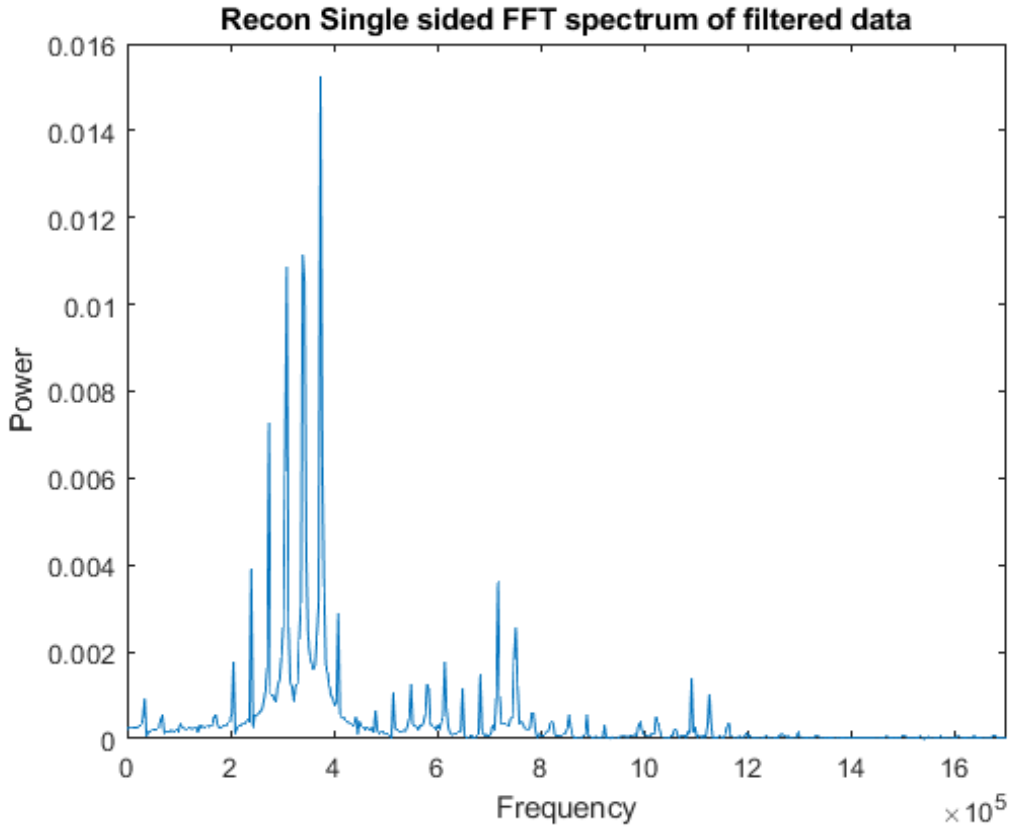


Figure 3.19: *FFT after SVD analysis of the signal from the plasma system*

log-log plot of FFT after SVD analysis:

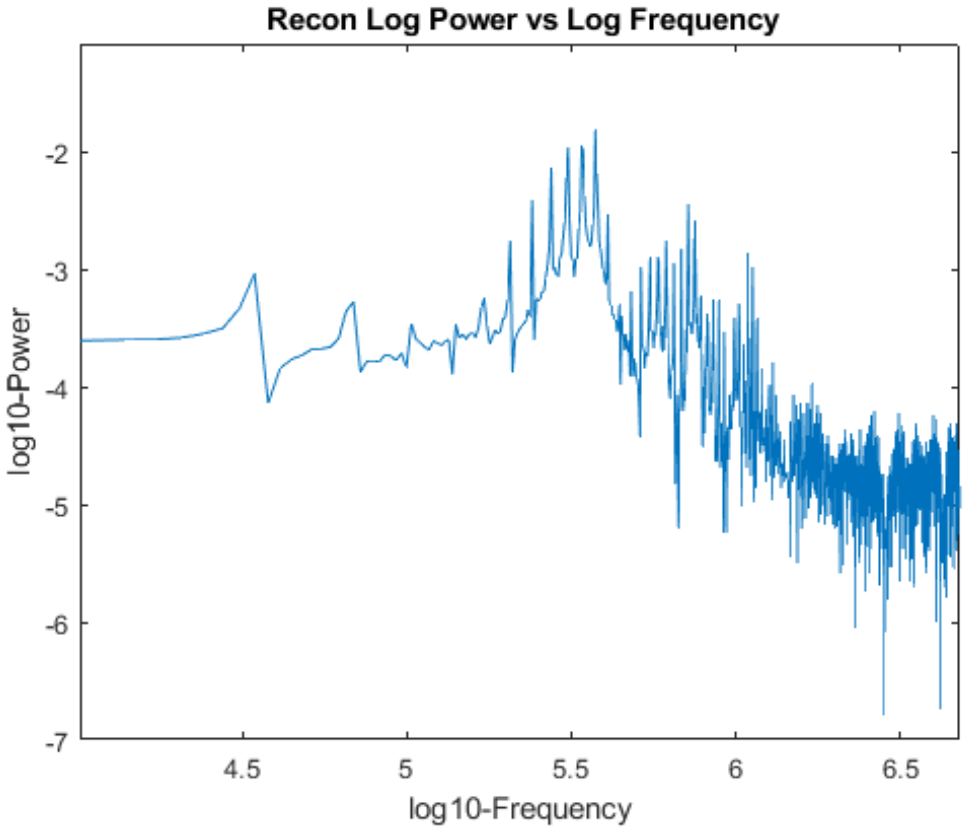


Figure 3.20: *log-log FFT after SVD analysis of the signal from the plasma system*

Reconstructed time series after applying butterworth filter over the SVD analysed time series:

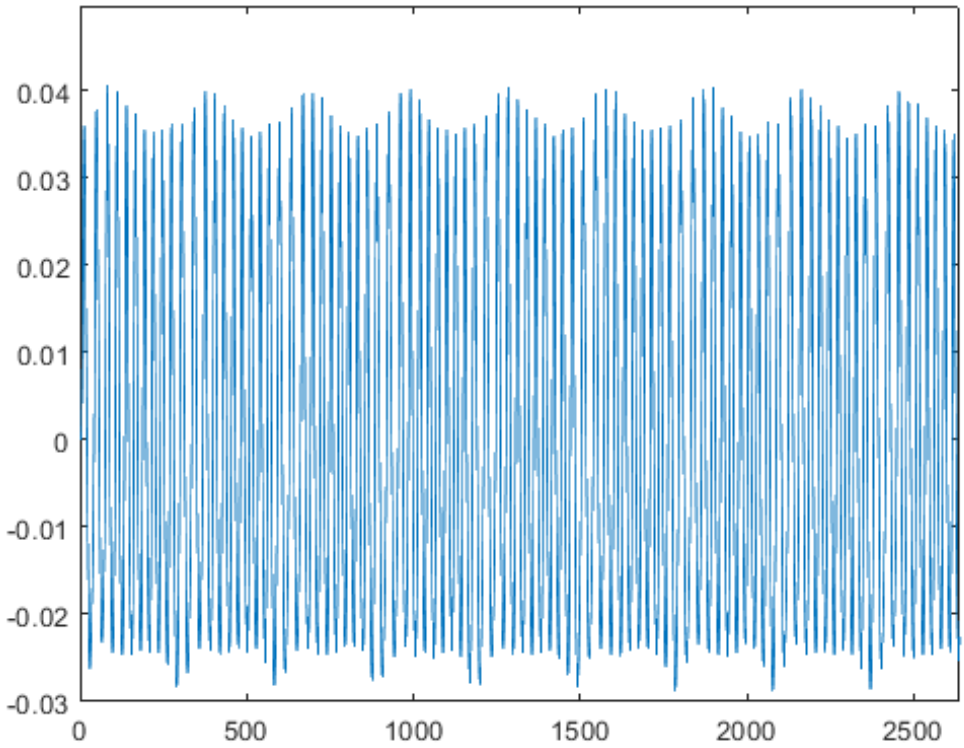


Figure 3.21: *Reconstructed time series after applying butterworth filter over the SVD analysed time series*

FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series:

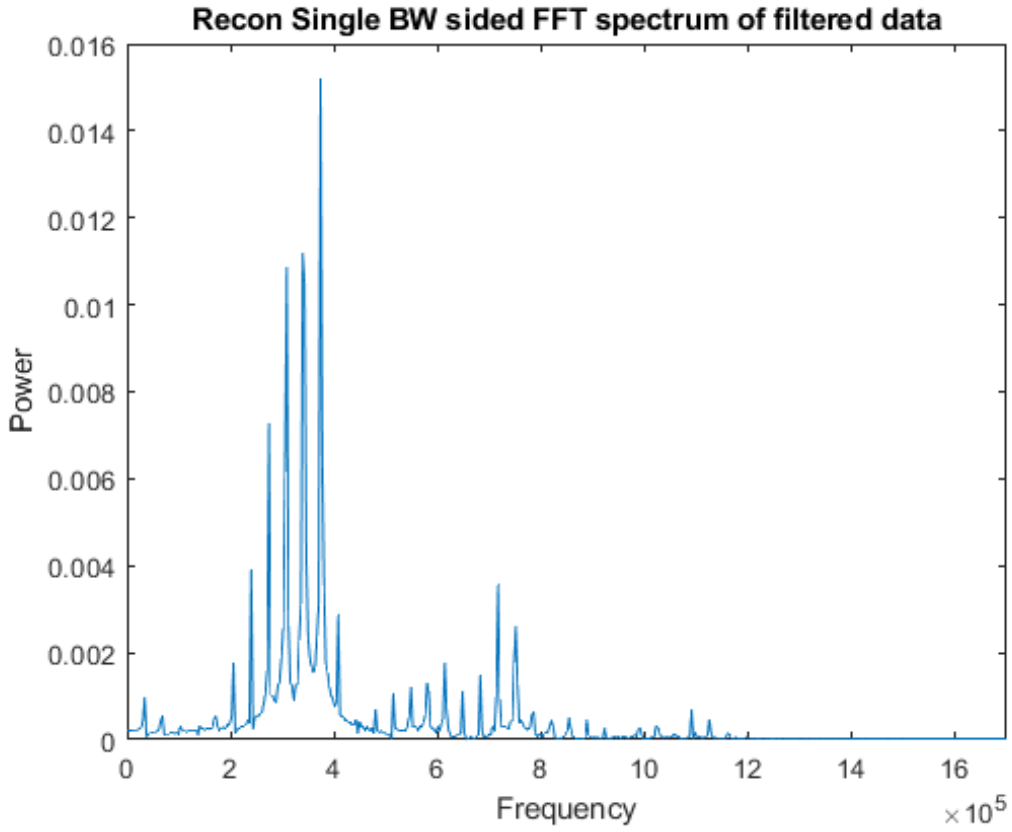


Figure 3.22: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

log-log FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series:

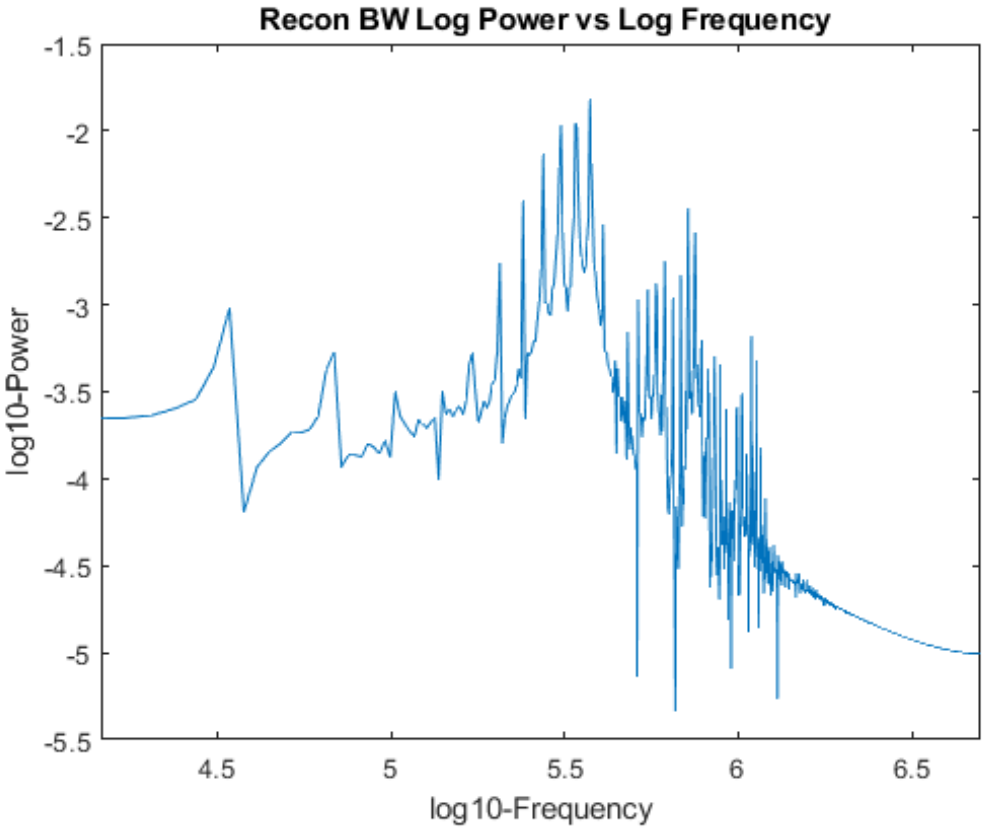


Figure 3.23: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

Phase portrait of reconstructed time series after applying butterworth filter over the SVD analysed time series:

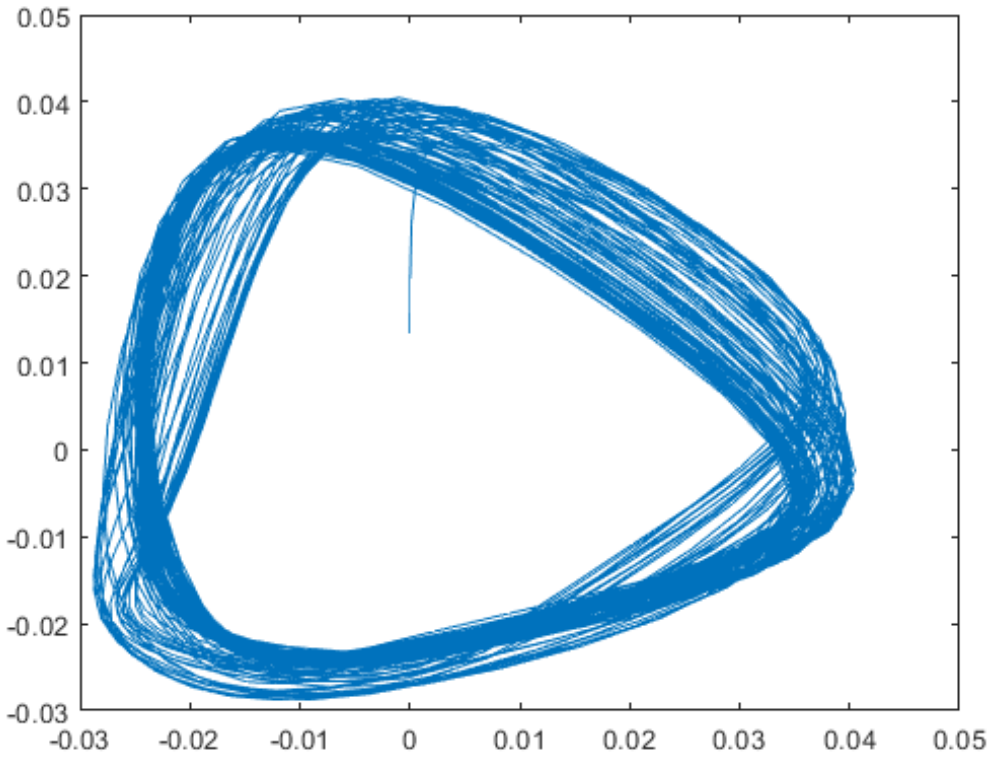


Figure 3.24: *Phase portrait of reconstructed time series after applying butterworth filter over the SVD analysed time series*

SVD analysis for embedding dimension (d)=5:

FFT after SVD analysis:

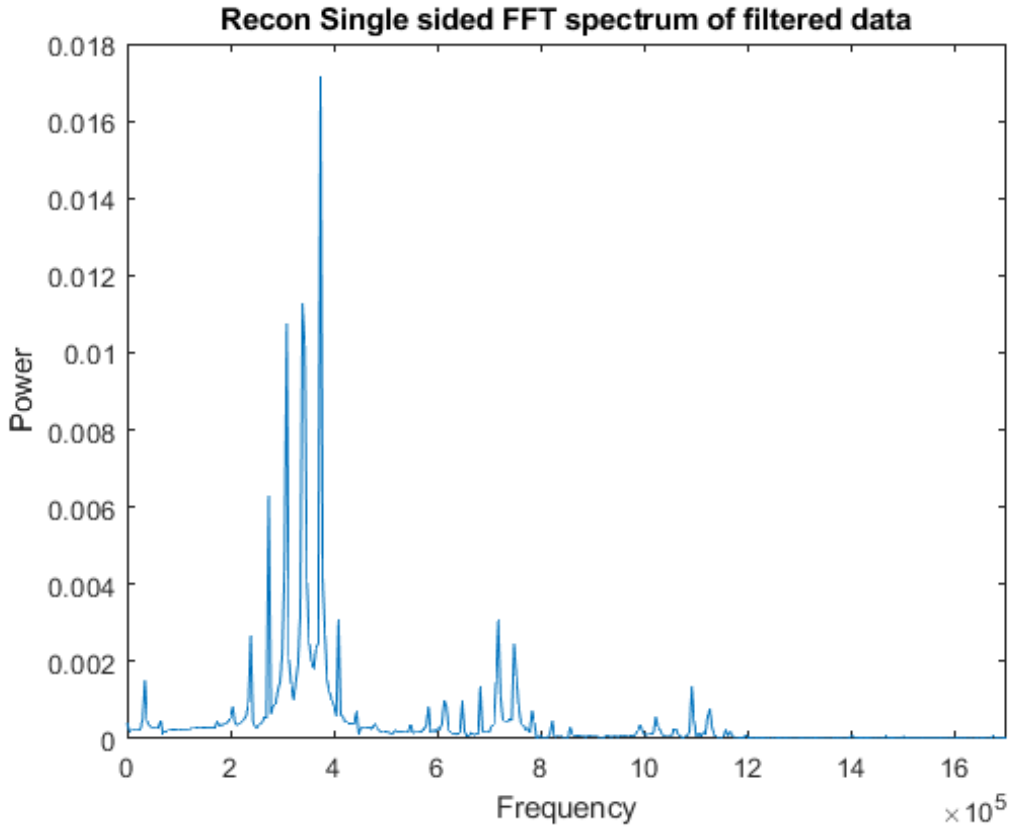


Figure 3.25: *FFT after SVD analysis of the signal from the plasma system*

log-log plot of FFT after SVD analysis:

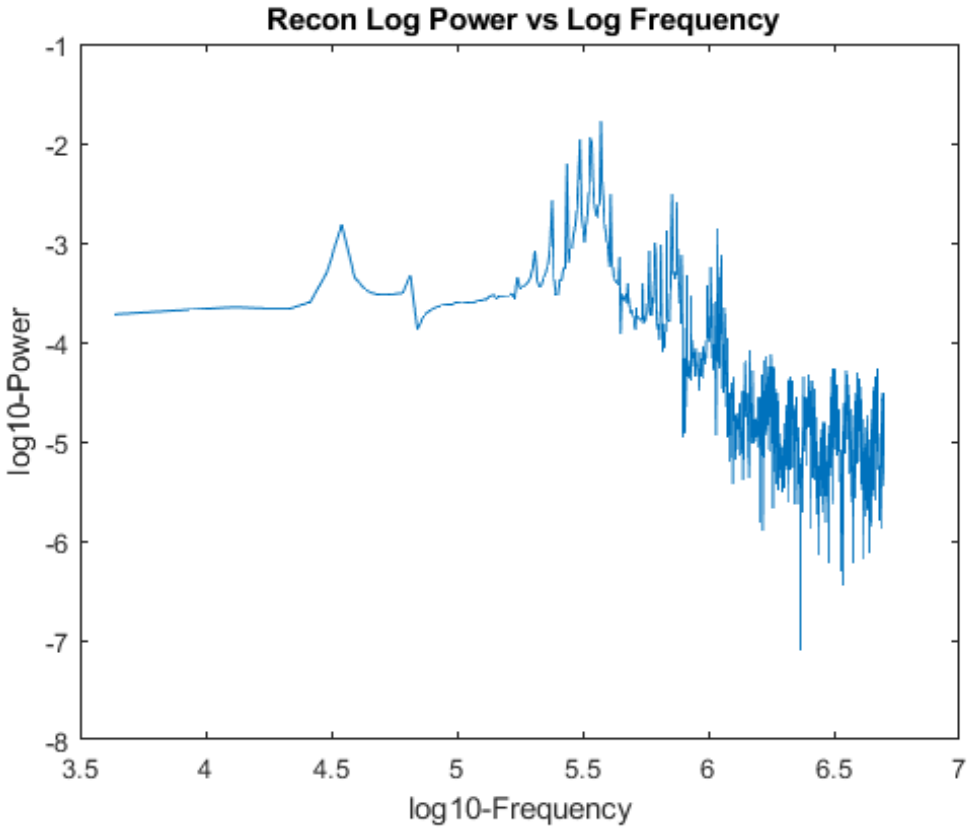


Figure 3.26: *log-log FFT after SVD analysis of the signal from the plasma system*

Reconstructed time series after applying butterworth filter over the SVD analysed time series:

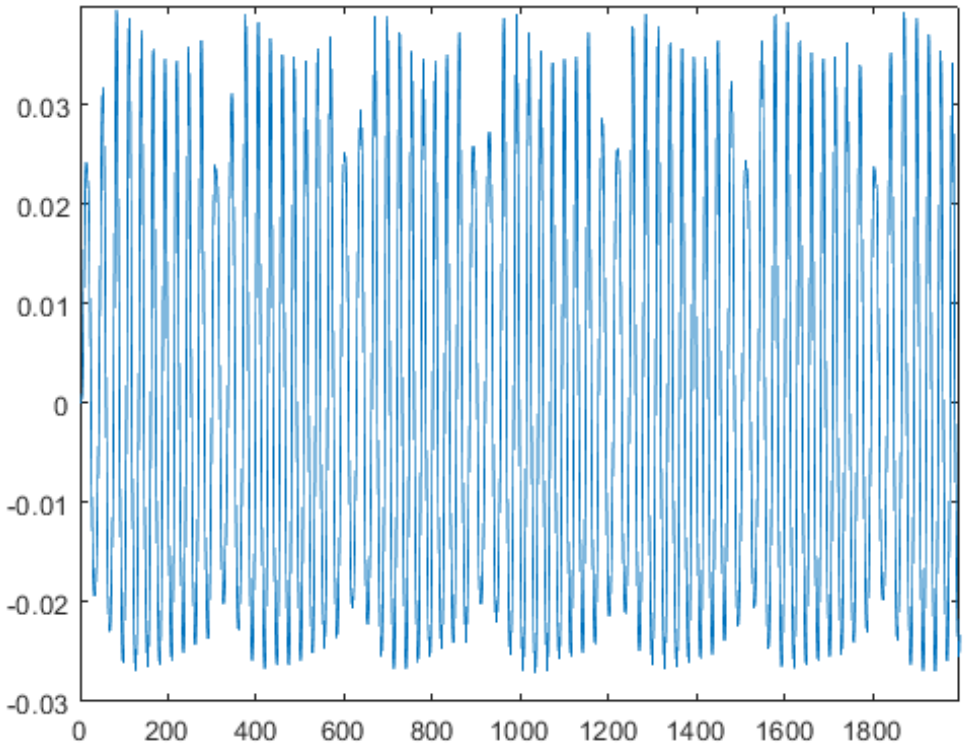


Figure 3.27: *Reconstructed time series after applying butterworth filter over the SVD analysed time series*

FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series:

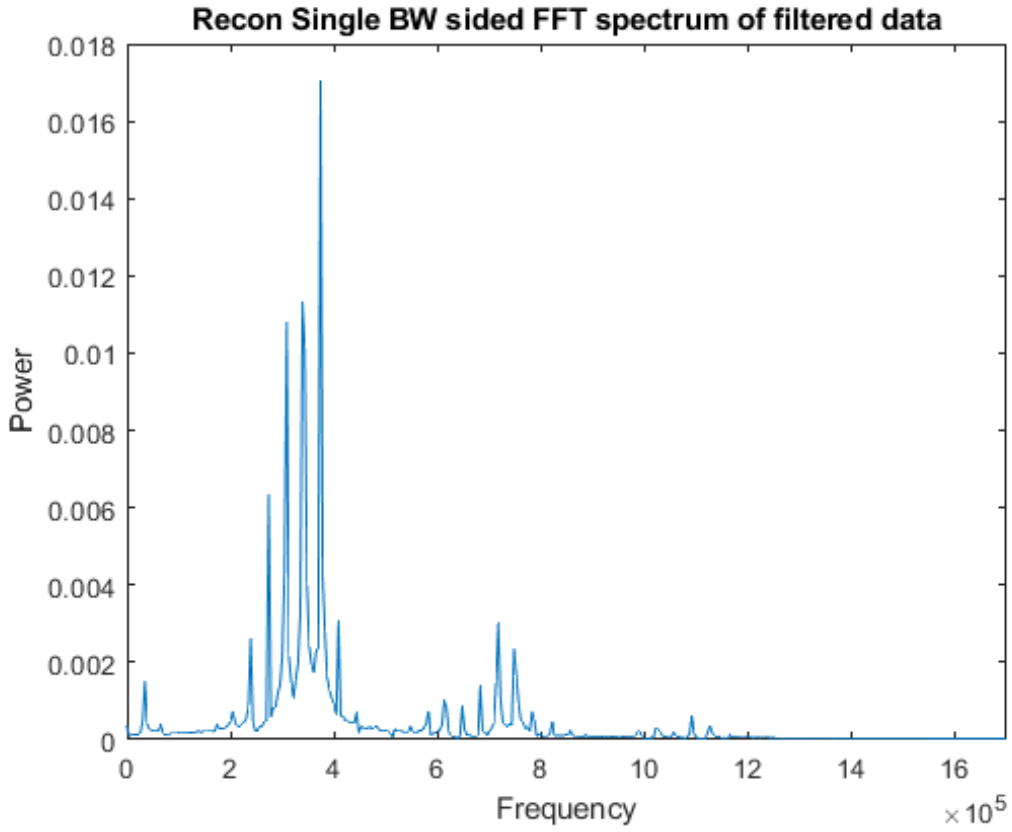


Figure 3.28: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

log-log FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series:

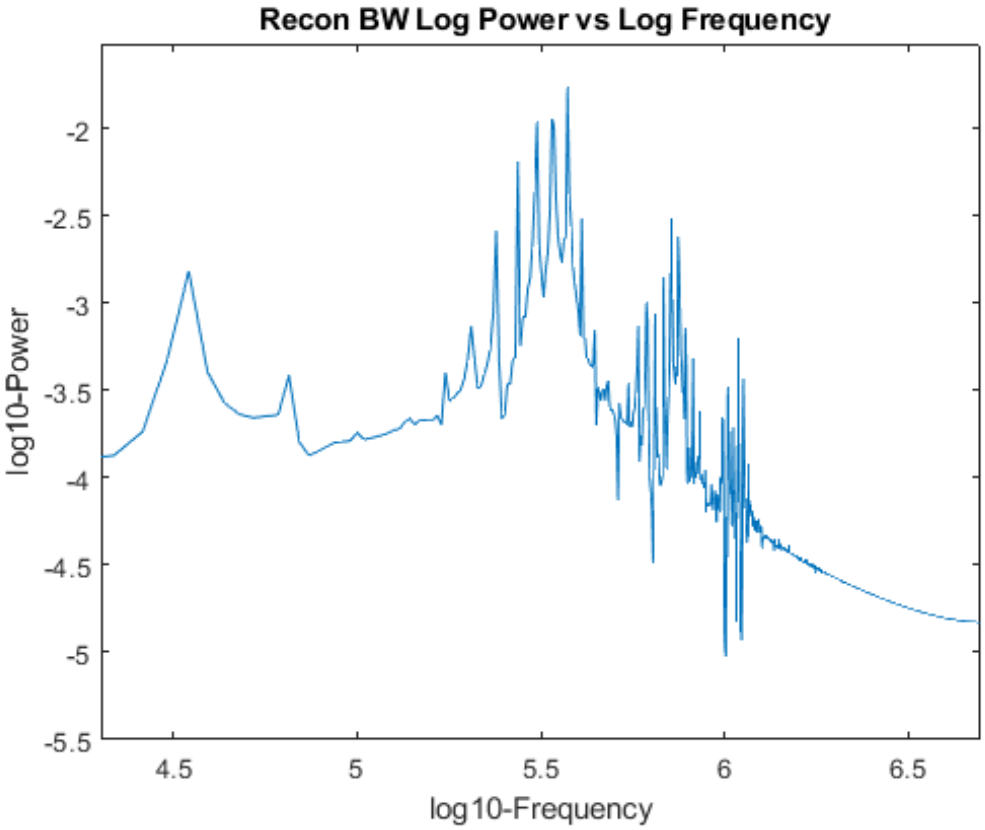


Figure 3.29: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

Phase portrait of reconstructed time series after applying butterworth filter over the SVD analysed time series:

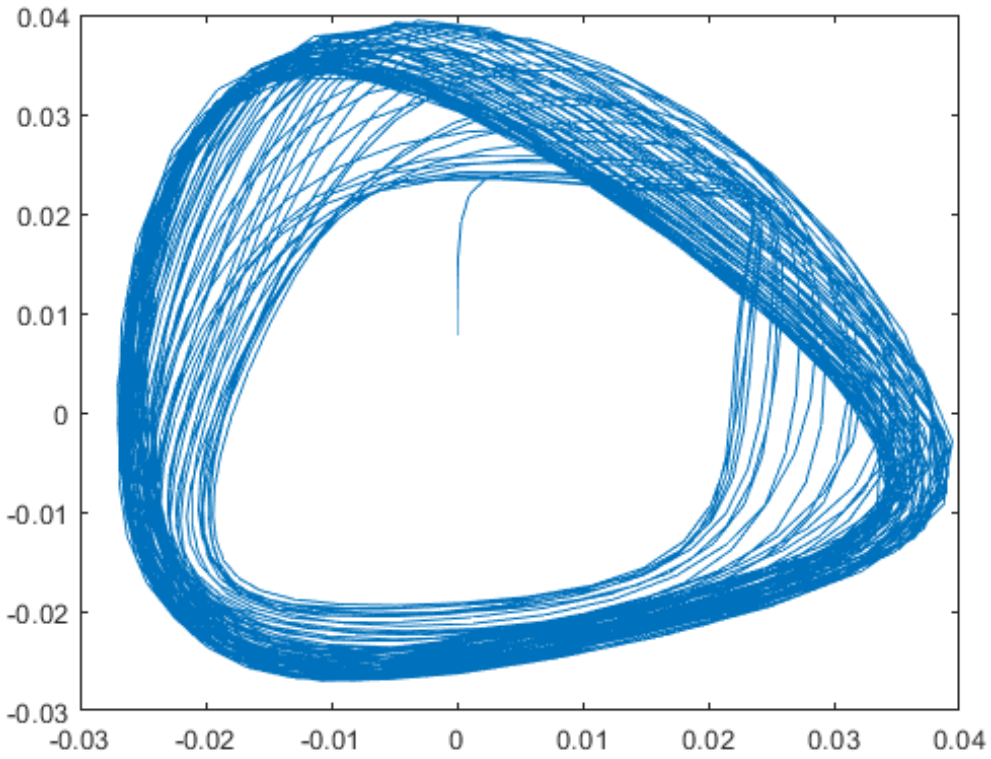


Figure 3.30: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

SVD analysis for embedding dimension (d)=7:

FFT after SVD analysis:

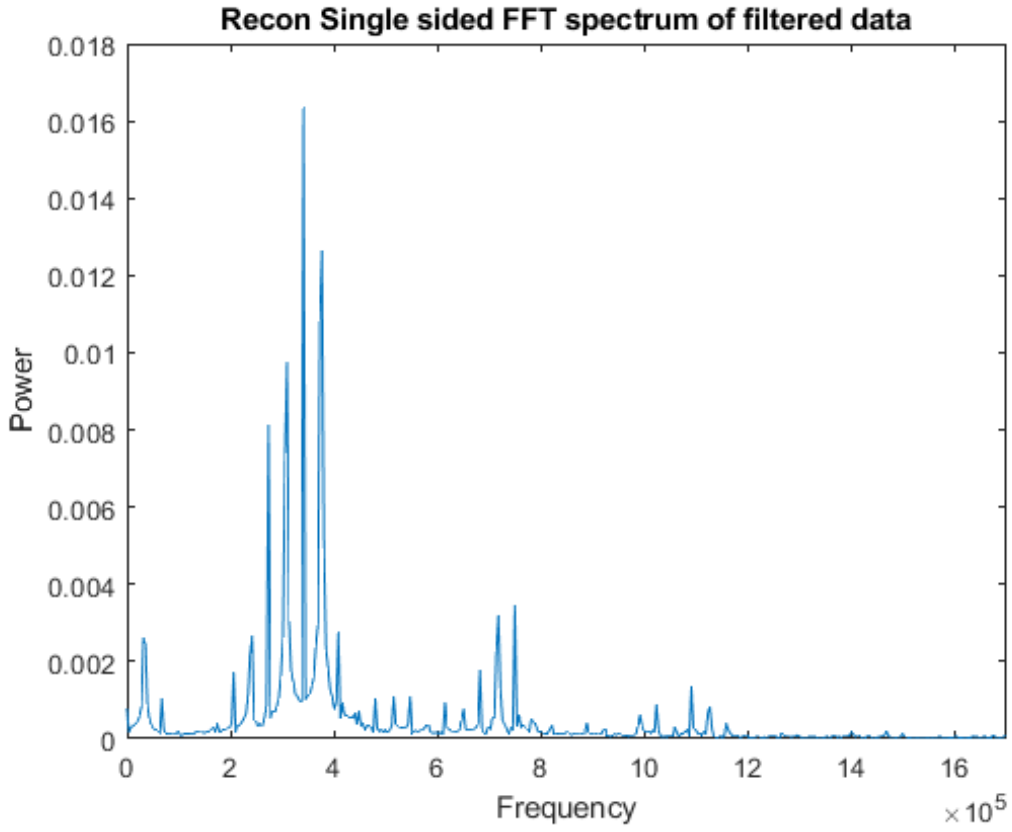


Figure 3.31: *FFT after SVD analysis of the signal from the plasma system*

log-log plot of FFT after SVD analysis:

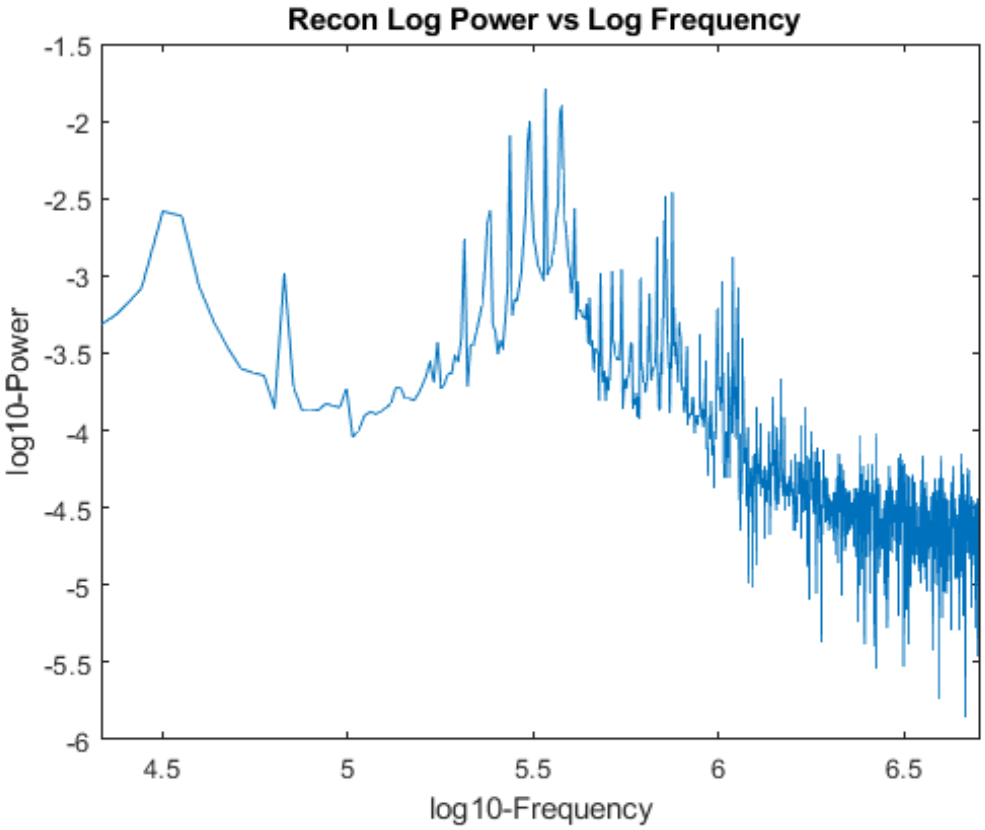


Figure 3.32: *log-log FFT after SVD analysis of the signal from the plasma system*

Reconstructed time series after applying butterworth filter over the SVD analysed time series:

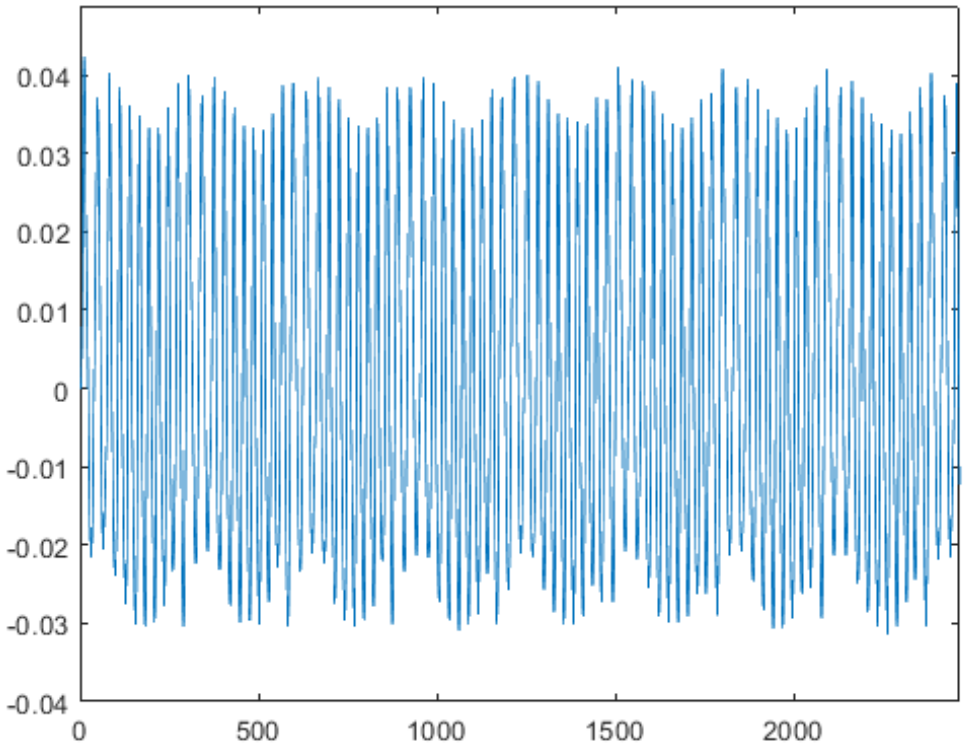


Figure 3.33: *Reconstructed time series after applying butterworth filter over the SVD analysed time series*

FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series:

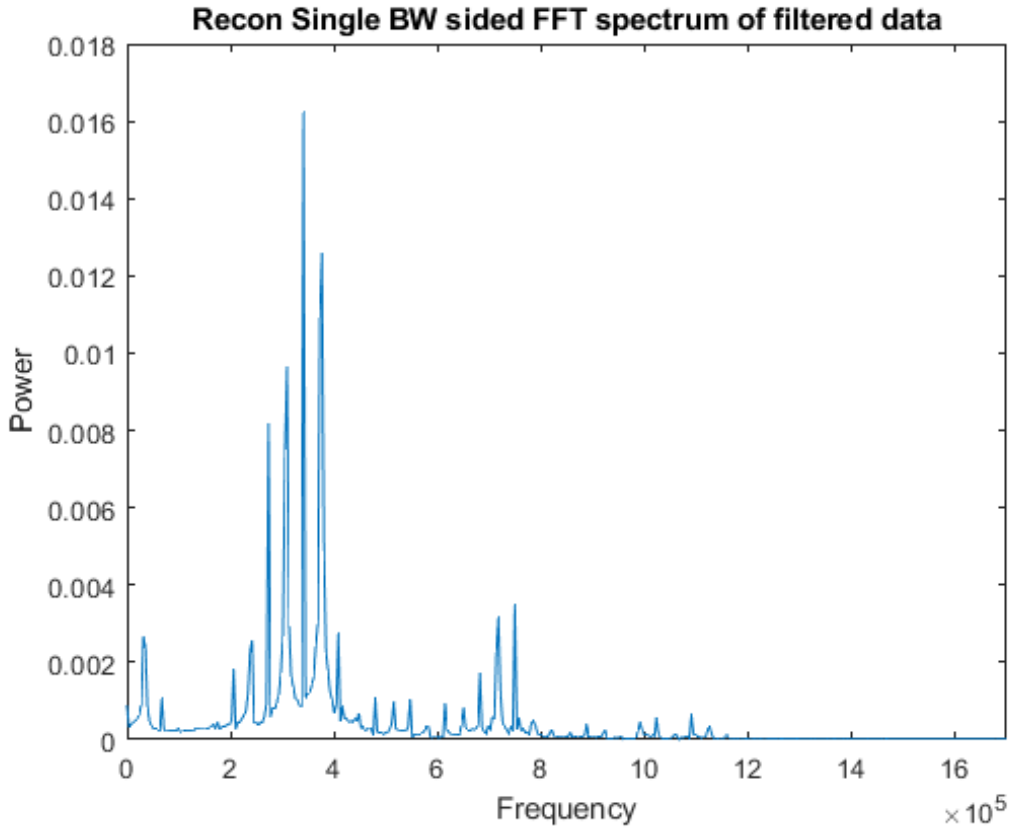


Figure 3.34: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

log-log FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series:

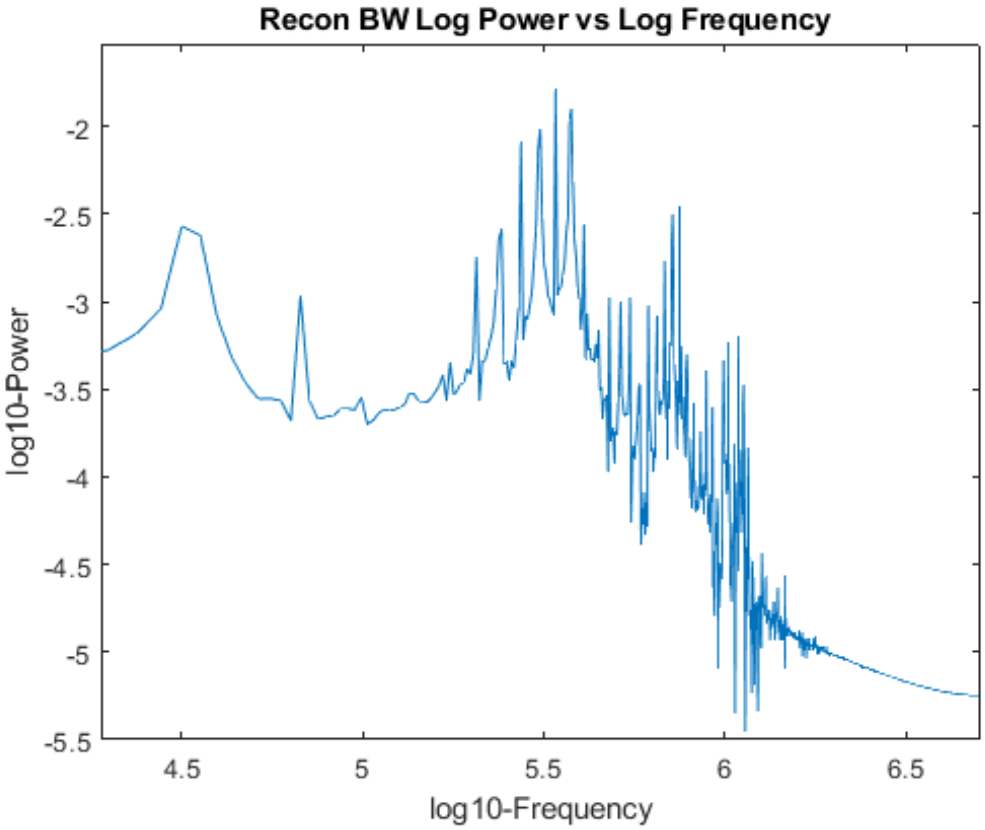


Figure 3.35: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

Phase portrait of reconstructed time series after applying butterworth filter over the SVD analysed time series:

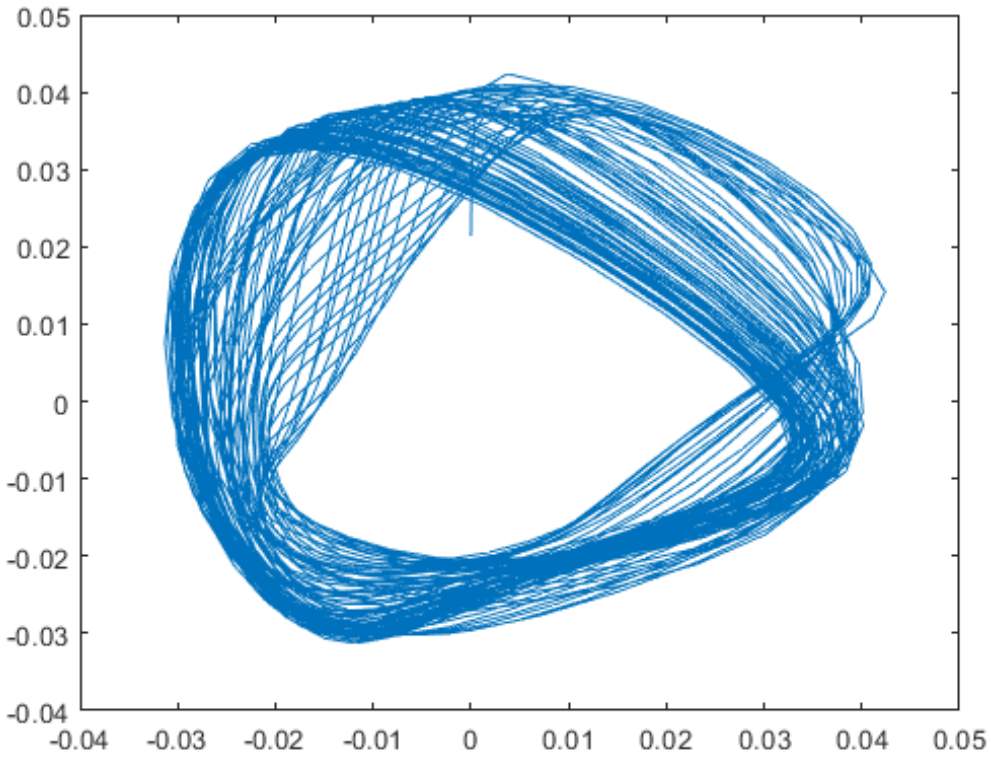


Figure 3.36: *FFT of reconstructed time series after applying butterworth filter over the SVD analysed time series*

In the above plots we can see that the noise is being filtered from the original time series if we compare figure 3.16 and figure 3.19 as is also seen comparing the log-log FFTs.

Additionally, a butterworth filter is also applied to the SVD reconstructed time series which further filters some amount of noise.

Also, we see that the best reconstruction of the phase portrait occurs when the embedding dimension is taken d=5, which is consistent with the embedding dimension found earlier.

Chapter 4

Conclusions

So far in this project, I have optimized the plasma production experimental technique and have obtained some of the plasma fluctuations. I have also performed the experiment for the Paschen's curve which is in accordance with the Paschen's Law.
Also, I have employed some non-linear analysis tools to investigate the underlying dynamics of such fluctuations, such as FFT and Phase Space reconstruction of the time series data from the plasma system along with the Recurrence Plots and determination of the Embedding Dimension.
We also saw the application of Singular Value Decomposition as a tool to filter the noise from the data.

Bibliography

1. Practical method for determining the minimum embedding dimension of a scalar time series Liangyue Cao Department of Mathematics, University of Western Australia, Nedlands, WA 6907, Australia

2. ITERATIVE SVD METHOD FOR NOISE REDUCTION OF LOW-DIMENSIONAL CHAOTIC TIME SERIES K.SHIN, J.K.HAMMOND AND P.R.WHITE ISVR, University of Southampton,Southampton SO17 IBJ, U.K.

3. SVD analysis of the magnetospheric AE index time series and comparison with low-dimensional chaotic dynamics M. A. Athanasiu, G. P. Pavlos

4. Francis F. Chen: Introduction to Plasma Physics and Controlled Fusion. Springer International Publishing, 2016.

5.Steven H. Strogatz: Nonlinear dynamics and Chaos: with application to physics, biology, chemistry, and Engineering. Perseus Books, Massachusetts, 2001.

6. F. Takens: Detecting Strange Attractors in Turbulence. Lecture Notes in Math. Vol. 898, Springer, New York (1981).

Appendices

Appendix A

MATLAB Codes

A.1 1. Fast Fourier Transform (FFT) of the Time series

Sine Wave

```
time = 0:0.001:1;

volt = sin(2*pi*50*time);
%volt = sawtooth(2*pi*100*time);


subplot(211);
plot(time,volt);
xlim([0  0.25])
xlabel('Time');
ylabel('Voltage');
title('Time series');

% FFT
% extract sampling information
samplingTime = time(2) - time(1);
samplingFreq = 1.0/samplingTime;
sampleLength = length(time);

% do fft and compute two sided spectrum
fftVolt = fft(volt);
absFftVolt = abs(fftVolt/sampleLength);

% prepare single sided spectrum
frequency = samplingFreq*(0:(sampleLength/2))/sampleLength;
power = 2*absFftVolt(1:(sampleLength/2)+1);

subplot(212);
```

```
plot(frequency, power);
xlim([0 1000]);
xlabel('Frequency');
ylabel('Power');
title('Single sided FFT spectrum of data');
```

Sine wave super position:

```
time = 0:0.001:1;
volt = sin(2*pi*100*time)+sin(2*pi*200*time);

subplot(211);
plot(time,volt);
xlim([0 0.25])
xlabel('Time');
ylabel('Voltage');
title('Time series');

% FFT
% extract sampling information
samplingTime = time(2) - time(1);
samplingFreq = 1.0/samplingTime;
sampleLength = length(time);

% do fft and compute two sided spectrum
fftVolt = fft(volt);
absFftVolt = abs(fftVolt/sampleLength);

% prepare single sided spectrum
frequency = samplingFreq*(0:(sampleLength/2))/sampleLength;
power = 2*absFftVolt(1:(sampleLength/2)+1);

subplot(212);
plot(frequency, power);
xlim([0 1000]);
xlabel('Frequency');
ylabel('Power');
title('Single sided FFT spectrum of data');
```

Non-linear Signal

```
format long
i1 = sprintf("%04d",53);
file = ""+"tek"+i1+"CH2";
csv = ""+file+".csv";


% Reading Data from file
data = csvread(csv, 21, 0);
volt = data(:,2);
time = data(:,1);
%Creating offsets for time as it is taken from oscilloscope
offset = -time(1);
time = time + time(1);

figure('numbertitle','off','Name','Time Series');
plot(time,volt);

% FFT
% extract sampling information
samplingTime = time(2) - time(1);
samplingFreq = 1.0/samplingTime;
sampleLength = length(time);

% using Butterworth Lowpass filter to filter the noise
 cutoff_freq = 1e+5;
% [b,a] = butter(6,cutoff_freq/(samplingFreq/2));
% volt = filter(b,a,volt);


% do fft and compute two sided spectrum
fftVolt = fft(volt);
absFftVolt = abs(fftVolt/sampleLength);

% prepare single sided spectrum
frequency = samplingFreq*(0:(sampleLength/2))/sampleLength;
power = 2*absFftVolt(1:(sampleLength/2)+1);

figure('numbertitle','off','Name','Frequency');
plot(frequency, power);
xlim([0,cutoff_freq*1.5]);
xlabel('Frequency');
ylabel('Power');
title('Single sided FFT spectrum of filtered data');
```

A.2 Phase Portrait

A.2.1 Phase Portrait of various signals

Sine Wave

```
time = 0:0.0001:1;
volt = sin(2*pi*50*time);
subplot(121);
plot(time,volt);
%xlim([0 0.0025])
xlabel('Time');
ylabel('Voltage');
title('Time series');
acf = autocorr(volt, floor(length(volt)/4));

 acf = abs(acf);

 i = 1; % iteration counter

 tol = 0.05; %tolerance value

 while(acf(i) >= tol)

     i = i + 1;

 end

 tau = i - 1; % embedding delay parameter

x = volt(1:length(volt)-tau);
y = volt(1+tau:length(volt));
subplot(122)
plot(x,y);
```

Quasi-Periodic signal

```
time = 0:0.0001:2;
volt = sin(2*pi*50*time)+sin(2*pi*49*time);
subplot(121);
plot(time,volt);
%xlim([0 0.0025])
xlabel('Time');
```

```
ylabel('Voltage');
title('Time series');
acf = autocorr(volt, floor(length(volt)/4));

 acf = abs(acf);

 i = 1; % iteration counter

 tol = 0.05; %tolerance value

 while(acf(i) >= tol)

     i = i + 1;

 end

 tau = i - 1; % embedding delay parameter

x = volt(1:length(volt)-tau);
y = volt(1+tau:length(volt));
%figure('numbertitle','off','Name','Phase Portrait');
subplot(122)
plot(x,y);
```

Quasi-Periodic signal

```
 time = 0:0.0001:2;
volt = sin(2*pi*50*time)+sin(2*pi*49*time);
subplot(121);
plot(time,volt);
%xlim([0 0.0025])
xlabel('Time');
ylabel('Voltage');
title('Time series');
acf = autocorr(volt, floor(length(volt)/4));

 acf = abs(acf);

 i = 1; % iteration counter

 tol = 0.05; %tolerance value

 while(acf(i) >= tol)

     i = i + 1;
```

```
 end

 tau = i - 1; % embedding delay parameter

x = volt(1:length(volt)-tau);
y = volt(1+tau:length(volt));

subplot(122)
plot(x,y);
```

Periodic signal with random noise

```
time = 0:0.0001:1;
volt = sin(2*pi*10*time)+rand(1,length(time));


subplot(121);
plot(time,volt);
xlabel('Time');
ylabel('Voltage');
title('Time series');
acf = autocorr(volt, floor(length(volt)/4));

 acf = abs(acf);

 i = 1; % iteration counter

 tol = 0.05; %tolerance value

 while(acf(i) >= tol)

     i = i + 1;

 end

 tau = i - 1; % embedding delay parameter

x = volt(1:length(volt)-tau);
y = volt(1+tau:length(volt));

subplot(122)
plot(x,y);
```

Non-Linear Signal

```
i1 = sprintf("%04d",53);
file = ""+"tek"+i1+"CH2";
csv = ""+file+".csv";


% Reading Data from file
data = csvread(csv, 21, 0);

volt = data(:,2);
time = data(:,1);
%Creating offsets for time as it is taken from oscilloscope


% Normalising the data
 volt = volt - mean(volt);

subplot(121);
plot(time,volt);

xlabel('Time');
ylabel('Voltage');
title('Time series');
acf = autocorr(volt, floor(length(volt)/4));

 acf = abs(acf);

 i = 1; % iteration counter

 tol = 0.05; %tolerance value

 while(acf(i) >= tol)

     i = i + 1;

 end

 tau = i - 1; % embedding delay parameter

x = volt(1:length(volt)-tau);
y = volt(1+tau:length(volt));

subplot(122)
plot(x,y);
```

A.2.2 Optimal Embedding Dimension

The code below is a function and not a standalone code.

```
function [tau, d_embed] = cao_embed(sig, max_d)

    % If maximum dimension is not given, default = 10 %

    if nargin < 2

        max_d = 10;

    end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% If signal length is greater than 5000, reduce it to length ~5000 %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    if length(sig) > 2500

        factor = floor(length(sig)/2500);
        %factor = floor(length(sig)/floor_alph);

    end

    sig = sig(1:factor:end);


    N = length(sig); % length of the new signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%          calculate the delay from autocorrelation function          %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    acf = autocorr(sig, floor(length(sig)/4));
```

```
    acf = abs(acf);

    i = 1; % iteration counter

    tol = 0.05; %tolerance value

    while(acf(i) >= tol)

        i = i + 1;

    end

    tau = i - 1; % embedding delay parameter


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%                         Generate a delay matrix                          %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



    min_d = 2; % set minimum dimension to 2


    % create a zero matrix of [row col]
    delay_matrix = zeros([N-(min_d-1)*tau max_d]);

    start_row = 1;

    for dim = max_d:-1:min_d

        for i = start_row:(N-(dim-1)*tau)

            index = i;


            for j = 1:dim


                if index <= N
```

```
                    delay_matrix(i, j) = sig(index);

                else

                    delay_matrix(i, j) = sig(index - N);

                end

                index = index + tau;

            end

        end

        start_row = N-(dim-1)*tau + 1;

    end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  Calculate the optimum E1 value

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    a_sum = zeros([max_d 1]);

    E = zeros([max_d 1]);

    E_star = zeros([max_d 1]);

    % reason: for a max dimension 10 it will calculate E1 upto 8
    E1 = zeros([max_d-2 1]);

    E2 = zeros([max_d-2 1]);

for dim = min_d:max_d-1

    for i = 1:1:N-dim*tau

    nearest_distance = 100; % some initial large value
% each delay vector initialized with nearest neighbor
%index same as itself
    nn_index = i;
```

```
for j = 1:1:N-dim*tau

distance=max(abs(delay_matrix(i, 1:dim) - delay_matrix(j, 1:dim)));

    if distance < nearest_distance && distance ~= 0

        nearest_distance = distance;

            nn_index = j;

                end

            end

max_norm_dim = max(abs(delay_matrix(i, 1:dim)
- delay_matrix(nn_index, 1:dim)));

max_norm_dim_plus = max(abs(delay_matrix(i, 1:dim+1)
- delay_matrix(nn_index, 1:dim+1)));

a_sum(dim) = a_sum(dim) + max_norm_dim_plus/max_norm_dim;



    % Calculation for E*

E_star(dim) = E_star(dim) + abs(sig(i + dim*tau)
- sig(nn_index + dim*tau));

        end


        E(dim) = 1/(N-dim*tau) * a_sum(dim);

        E_star(dim) = 1/(N-dim*tau) * E_star(dim);

        if dim > min_d

           E1(dim-1) = E(dim)/E(dim-1);

           E2(dim-1) = E_star(dim)/E_star(dim-1);

        end
```

```
        %subplot(221)

        %subplot(224)

    end
    xlabel('d');
    ylabel('E1&E2');
    plot(E1,'-o')

    hold on;

    plot(E2,'-*')

    hold off;


    E2_tol = 0.05;

    d_embed = max_d; % initial guess

    for dim = (1:max_d-3)

        if abs(E2(dim+1) - E2(dim)) <= E2_tol

            d_embed = dim;

            break;

        end

    end

end
```

A.3 Recurrence Plots

```
fig = figure('Position',[1 1 1536 788.8000]);
    i1 = sprintf("%04d",1);
    file = ""+"tek"+i1+"CH2";
    csv = ""+file+".csv";
    data = csvread(csv,21,0);
    volt = data(:,2);
    time = data(:,1);

    if length(volt) > 2500

            factor = floor(length(volt)/5000);


        end

        volt = volt(1:factor:end);



        N = length(volt); % length of the new signal

    %compensate for the offsets

    volt = volt-mean(volt);
    time = time - time(1);

    % extract sampling information
    samplingTime = time(2) - time(1);
    samplingFreq = 1.0/samplingTime;
    sampleLength = length(time);

    %using Butterworth Lowpass filter to filter the noise
    cutoff_freq = 1e5;
    [b,a] = butter(6,cutoff_freq/(samplingFreq/2));
    volt = filter(b,a,volt);
    volt=volt';

    %find optimum delay
    covar_mat = cov(volt);
    tra = trace(covar_mat);
    e_op = 0.2*sqrt(tra);
```

```
%Construction of delay matrix
del_mat = [volt(:,1) volt(:,1)];
recur_mat = zeros(N,N);
e = e_op;
for i = 1:N
    for j = 1:N
        if abs(volt(i)-volt(j))<=e
            recur_mat(i,j) = 1;
        else
            recur_mat(i,j) = 0;
        end
    end
end
imagesc(recur_mat);
```

A.4 Filtering

A.4.1 Singular Value Decomposition(SVD)

A.4.2 Main Code

```
format long
i1 = sprintf("%04d",53);
file = ""+"tek"+i1+"CH2";
csv = ""+file+".csv";


% Reading Data from file
data = csvread(csv, 21, 0);

volt = data(:,2);
time = data(:,1);

%Creating offsets for time as it is taken from oscilloscope
offset = -time(1);
time = time + time(1);



% Normalising the data
 volt = volt - mean(volt);


figure('numbertitle','off','Name','Time Series');
plot(time,volt);



 acf = autocorr(volt, floor(length(volt)/4));

 acf = abs(acf);

 i = 1; % iteration counter

 tol = 0.05; %tolerance value

 while(acf(i) >= tol)

     i = i + 1;
```

```
 end

 tau = i - 1; % embedding delay parameter



% FFT
% extract sampling information
samplingTime = time(2) - time(1);
samplingFreq = 1.0/samplingTime;
sampleLength = length(time);


 cutoff_freq = 1e+5;



% do fft and compute two sided spectrum
fftVolt = fft(volt);
absFftVolt = abs(fftVolt/sampleLength);

% prepare single sided spectrum
frequency = samplingFreq*(0:(sampleLength/2))/sampleLength;
power = 2*absFftVolt(1:(sampleLength/2)+1);

figure('numbertitle','off','Name','Frequency');
plot(frequency, power);
xlim([0,cutoff_freq*1.5]);
xlabel('Frequency');
ylabel('Power');
title('Single sided FFT spectrum of filtered data');

figure('numbertitle','off','Name','Log P log F');
plot(log10(frequency),log10(power));
xlabel('log10-Frequency');
ylabel('log10-Power');
title('Log Power vs Log Frequency');

x = volt(1:length(volt)-tau);
y = volt(1+tau:length(volt));
h1 = figure('numbertitle','off','Name','Phase Portrait');
plot(x,y);
```

```
     if length(volt) > 2500

                factor = floor(length(volt)/5000);
                volt = volt(1:factor:end);

    end

     [recon_tseries] = singular_decomposition(volt,time,11);


sampleLength = length(recon_tseries);

% do fft and compute two sided spectrum
fftVolt = fft(recon_tseries);
absFftVolt = abs(fftVolt/sampleLength);

% prepare single sided spectrum
frequency = samplingFreq*(0:(sampleLength/2))/sampleLength;
sampleLength;
power = 2*absFftVolt(1:(sampleLength/2)+1);

figure('numbertitle','off','Name','Recon Frequency');
plot(frequency, power);
xlim([0,10^6*1.7]);
xlabel('Frequency');
ylabel('Power');
title('Recon Single sided FFT spectrum of filtered data');

figure('numbertitle','off','Name','Recon Log P log F');
plot(log10(frequency),log10(power));
xlabel('log10-Frequency');
ylabel('log10-Power');
title(' Recon Log Power vs Log Frequency');




 % using Butterworth Lowpass filter to filter the noise
 cutoff_freq = 1e+6;
 [b,a] = butter(6,cutoff_freq/(samplingFreq/2));
 recon_tseries = filter(b,a,recon_tseries);

 figure('numbertitle','off','name','Reconstructed BW filtered time Series');
 plot(recon_tseries);
```

```
sampleLength = length(recon_tseries);
% do fft and compute two sided spectrum
fftVolt = fft(recon_tseries);
absFftVolt = abs(fftVolt/sampleLength);

% prepare single sided spectrum
frequency = samplingFreq*(0:(sampleLength/2))/sampleLength;
sampleLength;
power = 2*absFftVolt(1:(sampleLength/2)+1);

figure('numbertitle','off','Name','Recon BW Frequency');
plot(frequency, power);
xlim([0,cutoff_freq*1.7]);
xlabel('Frequency');
ylabel('Power');
title('Recon Single BW sided FFT spectrum of filtered data');


figure('numbertitle','off','Name','Recon BW Log P log F');
plot(log10(frequency),log10(power));
xlabel('log10-Frequency');
ylabel('log10-Power');
title(' Recon BW Log Power vs Log Frequency');



acf = autocorr(recon_tseries, floor(length(recon_tseries)/2));
acf = abs(acf);
i = 1; % iteration counter
tol = 0.05; %tolerance value
while(acf(i) >= tol)
        i = i + 1;
end
tau = i - 1; % embedding delay parameter



x = recon_tseries(1:length(recon_tseries)-tau);
y = recon_tseries(1+tau:length(recon_tseries));

h=figure('numbertitle','off','Name','Reconstructed Phase Portrait');
plot(x,y);
```

A.4.3 SVD function

```
function [recon_tseries,new_dmat]=singular_decomposition(sig,time,dim)

    if length(sig) > 2500

                factor = floor(length(sig)/5000);
                sig = sig(1:factor:end);

    end


     N = length(sig); % length of the new signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%          calculate the delay from autocorrelation function          %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    acf = autocorr(sig, floor(length(sig)/4));

    acf = abs(acf);

    i = 1; % iteration counter

    tol = 0.05; %tolerance value

    while(acf(i) >= tol)

        i = i + 1;

    end

    tau = i - 1; % embedding delay parameter


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%                          Generate a delay matrix                        %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
d=dim;
delay_mat = zeros([N d]);

for i = 1:N

        index = i;

        for j = 1:d

            if index <= N

                delay_mat(i,j) = sig(index);

            else

                delay_mat(i,j) = sig(index - N);

            end

            index = index + tau;
        end
end

[U, S, V] = svd(delay_mat);


dmat = delay_mat*delay_mat';

rnk = rank(dmat);


singular_values = zeros(rnk,1);
index_singular = 1;
for i=1:rnk
    for j=1:rnk
        if(i==j)
            singular_values(index_singular) = S(i,j);
        end
    end
    index_singular = index_singular + 1;
end
index_number = (1:rnk);
figure('numbertitle','off','name','Normalised singular values')
plot(index_number,singular_values);
```

```
    derivative_min = 1;
    for i=1:rnk-1

        derivative = diff(singular_values(i:i+1));
        if((derivative) < derivative_min)
            derivative_min = derivative;
            threshold = i+1;
        end
    end

    S = zeros(size(S));

    in = 1;
    for i=1:threshold
        for j=1:threshold
            if(i==j)
                S(i,j) = singular_values(in);
            end
        end
    in = in+1;
    end

    threshold
    S(1:10,:)

dom = diag(S);
new_dmat=U(:,1:threshold)*diag(dom(1:threshold))*V(:,1:threshold)';

    recon_tseries = new_dmat(:,1);

     dimen = sprintf("%02d",d);

[pks,locs]=findpeaks(recon_tseries,time(1:length(recon_tseries)));
     pks2 = pks(pks>=0);
     m = mean(pks2)-0.15*mean(pks2);
     M = mean(pks2)+0.1*mean(pks2);
     for i=floor(length(pks2)/2):length(pks2)

         if(pks2(i) < m || pks2(i) > M )

             flag = pks2(i);
             break;
```

```
        else
            flag = 0;
        end
    end
    if(flag ~= 0)
        for i=1:length(pks)

            if(pks(i)==flag)
                flag2 = i;
                break;
            end
        end
        t = locs(flag2);
        for i=1:length(time)

            if(time(i)==t)
                N = i;
                break;
            end
        end

    end
    N = N-150;
    recon_tseries = recon_tseries(1:N);
     N = length(recon_tseries);
```