

**FAULT DETECTION IN DYNAMIC SYSTEMS USING THE
LARGEST LYAPUNOV EXPONENT**

A Thesis

by

YIFU SUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Mechanical Engineering

**FAULT DETECTION IN DYNAMIC SYSTEMS USING THE
LARGEST LYAPUNOV EXPONENT**

A Thesis

by

YIFU SUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Alexander G. Parlos
Committee Members,	Alan B. Palazzolo Jose Silva-Martinez
Head of Department,	Dennis O'Neal

May 2011

Major Subject: Mechanical Engineering

ABSTRACT

Fault Detection in Dynamic Systems Using the Largest Lyapunov Exponent.

(May 2011)

Yifu Sun, B.S, Beijing Institute of Technology

Chair of Advisory Committee: Dr. Alexander G. Parlos

A complete method for calculating the largest Lyapunov exponent is developed in this thesis. For phase space reconstruction, a time delay estimator based on the average mutual information is discussed first. Then, embedding dimension is evaluated according to the False Nearest Neighbors algorithm. To obtain the parameters of all of the sub-functions and their derivatives, a multilayer feedforward neural network is applied to the time series data, after the time delay and embedding dimension are fixed. The Lyapunov exponents can be estimated using the Jacobian matrix and the QR decomposition. The possible applications of this method are then explored for various chaotic systems. Finally, the method is applied to some real world data to demonstrate the general relationship between the onset and progression of faults and changes in the largest Lyapunov exponent of a nonlinear system.

To my beloved parents and wife

ACKNOWLEDGEMENTS

Foremost, I would like to express my gratitude to my advisor Dr. Alexander G. Parlos. Without his patience and guidance, I could not have finished this thesis work.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Alan B. Palazzolo and Dr. Jose Silva-Martinez for their encouragement, insightful comments, and hard questions.

Last but not the least, I would like to thank my family: my parents Dahong Sun and Hong Ren, my wife Chunliu Mao. They always support me through my life.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF FIGURES.....	ix
LIST OF TABLES	xii
 CHAPTER	
I INTRODUCTION.....	1
1.1 Research Motivation	1
1.2 Literature Review	2
1.3 Problem Statement	8
1.4 Research Contribution.....	9
1.5 Organization.....	9
II INTRODUCTION OF CHAOS AND LYAPUNOV EXPONENTS ..	11
2.1 Introduction	11
2.2 Description of Chaos	12
2.3 Categories of Dynamical Systems.....	14
2.4 Sensitive Dependence upon Initial Conditions	17
2.5 Indication of Chaos	18
2.6 Chapter Summary.....	21
III COMPUTING THE LYAPUNOV EXPONENTS FROM MEASUREMENTS	22
3.1 Introduction	22
3.2 Problems with Computing the Lyapunov Exponents.....	23
3.3 Phase Space Reconstruction.....	26

CHAPTER	Page
3.3.1 Average Mutual Information.....	27
3.3.2 False Nearest Neighbors Method	29
3.4 Jacobian Matrix and the QR Decomposition	30
3.5 Artificial Neural Networks.....	35
3.5.1 Introduction of Artificial Neural Networks.....	35
3.5.2 Neural Network Learning.....	37
3.5.3 Multilayer Feedforward Networks	40
3.6 Chapter Summary.....	44
 IV SIMULATION RESULTS.....	 45
4.1 Introduction	45
4.2 The Lorenz Attractor	45
4.2.1 Introduction of the Lorenz Attractor and Its Phase Space Reconstruction.....	 45
4.2.2 The Largest Lyapunov Exponent Calculation for the Lorenz System	 49
4.2.3 Chaotic Dynamics of the Lorenz System	64
4.3 The Hénon Map.....	69
4.4 The Rössler Attractor	71
4.5 Chapter Summary.....	75
 V APPLICATION OF LARGEST LYAPUNOV EXPONENT TO FAULT DETECTION.....	 76
5.1 Introduction	76
5.2 Proposed Method for Incipient Fault Detection.....	76
5.3 A Real World Example	77
5.4 Experimental Results.....	79
5.5 Chapter Summary.....	85
 VI SUMMARY AND CONCLUSION.....	 86
6.1 Summary	86
6.2 Conclusion.....	87
6.3 Limitation and Future work.....	87
 REFERENCES.....	 89
 APPENDIX A	 94

APPENDIX B 98

VITA 99

LIST OF FIGURES

FIGURE	Page
2.1 The oscillation of the water at the tip before and after the water drop.....	13
2.2 Sketch of magnet, superconductor, and elastic beam support and excitation apparatus.....	14
2.3 The real space and state space of a pendulum with and without friction	15
2.4 The appearance of a torus attractor	16
2.5 The phase space of a Van der Pol attractor	16
3.1 Flow chart for calculating Lyapunov exponents	25
3.2 Flow chart for distinguishing false neighbors and real neighbors.....	30
3.3 The structure of an artificial neuron.....	36
3.4 The structure of a multilayer neural network.....	36
3.5 A single neuron structure	37
3.6 Neuron activation functions	42
4.1 Time series plots for the three states of the Lorenz system	46
4.2 Three-dimensional plot of the Lorenz system.....	47
4.3 Average mutual information for the Lorenz system	48
4.4 The result of applying the False Nearest Neighbors method to the Lorenz system.....	49
4.5 Calculated value of the LLE for the Lorenz system.....	51
4.6 Relative error of the LLE for the Lorenz system	52
4.7 Standard deviation of the LLE for the Lorenz system	53

FIGURE	Page
4.8 Average LLE for the Lorenz system with different time delay	54
4.9 Relative error of the LLE for the Lorenz system with different time delay	54
4.10 Standard deviation of the LLE for the Lorenz system with different time delay	55
4.11 The theoretical value and the calculated value of the LLE for the 5% or 10% parameters changed Lorenz system without retraining the neural network.....	57
4.12 The theoretical value and the calculated value of the LLE with the neural network trained by 3% parameters changed Lorenz system data	58
4.13 The theoretical value and the calculated value of the LLE for 5% parameters changed Lorenz system with retraining neural network.....	59
4.14 The theoretical value and the calculated value of the LLE for 10% parameters changed Lorenz system with retraining neural network.....	59
4.15 The theoretical value and the numerical value of the LLEs for the Lorenz system added Sinusoidal function input with the amplitude of 10	61
4.16 The theoretical value and the numerical value of the LLEs for the Lorenz system added Sinusoidal function input with the amplitude of 5	62
4.17 The theoretical value and the numerical value of the LLEs for the Lorenz system added Sinusoidal function input with the amplitude of 1	62
4.18 The theoretical LLE for the original Lorenz system and the numerical LLEs for the noise-added Lorenz system.....	64
4.19 The 3-dimensional plot of the Lorenz system with the parameters (a) $\sigma=10$, $R=0.5$, $\beta=8/3$; (b) $\sigma=10$, $R=10$, $\beta=8/3$; (c) $\sigma=10$, $R=28$, $\beta=8/3$..	66
4.20 Largest Lyapunov exponent evaluation for (a) the non-noise Lorenz system; (b) the Lorenz system added the noise with SNR=20; (c) with SNR=13; (d) SNR=10	67
4.21 The appearance of the attractor of Hénon map	70

FIGURE	Page
4.22 The result of applying the False Nearest Neighbors method to the Hénon map.....	71
4.23 Illustration of the reinjection principle between the two branches of a Z-shaped slow manifold allowing (a) periodic relaxation oscillations in dimension two and (b) higher types of relaxation behavior in dimension three.....	72
4.24 The appearance of the Rössler attractor.....	73
4.25 Average mutual information of the Rössler attractor.....	74
4.26 The result of applying the False Nearest Neighbors method to the Rössler attractor.....	74
5.1 Steps used in the staged motor fault.....	77
5.2 The time series plot of the data from (a) data file #10598 (b) data file #57292.....	78
5.3 Average mutual information of the data from the file #10598.....	79
5.4 The result of applying the False Nearest Neighbors method to the time series data from the induction motor system.....	80
5.5 Comparison of the Lyapunov exponents for the file #10598 and the file #57292 for 20 seconds of the evaluation time.....	81
5.6 Changes of the LLE for the current I_a through the entire damage process	82
5.7 The average value of the percentage changes of the LLE with error bars for each damage level.....	83
5.8 The maximum value of the percentage changes of the LLE for each damage level.....	84
5.9 Comparison of the distributions for the healthy system and the faulty system.....	85

LIST OF TABLES

TABLE	Page
4.1 Comparison between the largest Lyapunov exponent evaluation results with 0.1 and 0.16 seconds time delays	55
4.2 Summary of the results for different time gaps between the data set for training the neural network and the data set for evaluating the LLE	56
4.3 Three-dimensional plots for Lorenz system with different Sinusoidal inputs	60
A.1 The LLEs for the Lorenz system with different time length	94
A.2 The LLEs for the 5% and 10% parameters changed Lorenz systems without retraining the neural network	94
A.3 The LLEs with the neural network trained by 3% parameters changed Lorenz system data	95
A.4 The LLEs with the neural network trained by 5% parameters changed Lorenz system data	95
A.5 The LLEs with the neural network trained by 10% parameters changed Lorenz system data	95
A.6 The largest Lyapunov exponent of the Lorenz system added Sinusoidal function input with the amplitude of 10	96
A.7 The largest Lyapunov exponent of the Lorenz system added Sinusoidal function input with the amplitude of 5	96
A.8 The largest Lyapunov exponent of the Lorenz system added Sinusoidal function input with the amplitude of 1	96
A.9 The largest Lyapunov exponents evaluated from noise-added system data with the neural network trained by non-noise Lorenz system data	97
A.10 The largest Lyapunov exponents evaluated from noise-added system data with retraining neural network	97

TABLE	Page
B.1 The largest Lyapunov exponents of current Ia for 5 different damage levels.....	98
B.2 The largest Lyapunov exponents of current Ib for 5 different damage levels.....	98
B.3 The largest Lyapunov exponents of current Ic for 5 different damage levels.....	98

CHAPTER I

INTRODUCTION

1.1 Research Motivation

These days, failures of machinery and industrial equipment often cause downtime, costly repairs, and possibly catastrophic events. Therefore the fault detection and diagnosis of dynamical systems has become both a useful and important field. Numerous books and papers are devoted to various approaches to fault detection and diagnosis. Common fault detection and diagnosis methods include the model-based method, the data-based method, and the knowledge-based method. But in most practice situations, an accurate model of the dynamical system in question is not available. Even worse, very often only little knowledge and information about the system is known, and only time-series experimental output data is available. As a result, the data-based or data driven method is often the more promising route. Besides, nonlinear dynamical systems sometimes exhibit chaotic behavior, and the Lyapunov exponent is a useful tool to distinguish and measure the extent of chaos. Previous studies on chaos and on the Lyapunov exponents have found applications to several fields such as turbulence, communication, heartbeats, and so on. However, little research has been done on the relationship between the behavior of Lyapunov exponents and fault detection.

1.2 Literature Review

In the history of the field of fault detection, many methods have been proposed to extract and analyze experimental data in order to detect faults and assist in diagnosis, including autoregressive modeling [1], empirical model decomposition [2], wavelet and wavelet packet methods [3], etc. However, in real world situations, the failures of systems such as electric power systems, manufacturing machines, and automatic control systems are always accompanied by nonlinear dynamics which may exhibit chaotic behavior, especially as motion changes from regular to chaotic. Many traditional methods can't effectively extract these useful nonlinear features. As a result, during last two decades researchers have begun to propose a wide variety of methods geared toward detecting faults in nonlinear dynamical systems, such as time-frequency analysis, and so on. However it is still difficult to detect faults early on due to certain weakly developing faults usually covered by background noise and other chaotic elements. If the signal to noise ratio (SNR) is low, weak fault signals may not be extracted from background noise using only the above-mentioned methods. In recent years, as chaos theory has developed, some new technologies (especially phase space reconstruction) have begun to be applied to extract information hidden beneath experimental data.

Chaos was first experimentally identified in weather prediction problems by Edward Lorenz in 1960 and the discovery was published in 1963[4]. For distinguishing whether or not chaos appears, the Lyapunov exponent is usually considered as a very important and useful indicator, especially with the development of the phase space reconstruction technique. According to Oseledec's fundamental paper [5], Lyapunov

exponents measure the exponential rates of divergence or convergence of nearby orbits of an attractor in the state space. A system with one or more positive Lyapunov exponents is defined as chaotic. An analysis of Lyapunov exponents can give researchers information about the extent of chaotic contribution in the system. Generally, Lyapunov exponents can be applied as features during the diagnostic process in order to differentiate one system state from another.

Wolf *et al* [6] presented the first algorithm to estimate non-negative Lyapunov exponents from an experimental time series data set; such a data set is usually available from observations. This algorithm examines the orbital divergence of length scales that are always as small as possible, using the Gram-Schmidt reorthonormalization (GSR) procedure to reconstruct the phase space of the system while the Lyapunov exponent is calculated by the phase space evolution.

In Wolf's method, the embedding dimension and the reconstruction time delay need to be provided or estimated. An unsuitable embedding dimension or time delay could cause an undesirable calculation result of the Lyapunov exponent. At the time this research was first published, a number of researchers focused on exploring methods to find the ideal embedding dimension and the time delay.

Rhodes [7] originally developed the algorithm called False Nearest Neighbors (FNN) to determine the embedding dimension of an autonomous time series. This method is a tool used to determine if one prescribed vector contains enough information to predict another vector directly, solely from the properties of the data. In this paper, the

author introduces the definition of the False Nearest Neighbor and the method by using the False Nearest Neighbor as a criterion.

Then Kennel and Abarbanel [8] demonstrate a more reliable method False Neighbors and False Strands, to estimate the minimum necessary embedding dimension. This research improves upon previous work by correcting for systematic effects. With greater cost of computation, the method can distinguish easily between colored noise and low-dimensional dynamics. In 2002, Min Lei *et al.* used another method, the Symplectic Geometry (SG) method [9], to estimate the embedding dimension. The SG method, which measures preserving characteristic and is capable of retaining unchanged the essential character of the primary time series while performing symplectic similar transformations. In addition, the SG method does not strongly depend upon the length of the time series, and is not impacted by either noise or sampling time.

On the one hand, the embedding dimension can be evaluated by many methods, especially FNN (which is the most well developed method). On the other hand, a lot of work has been done by researchers on the reconstruction delay time. In Michael T. Rosenstein's paper [10], the author mentioned that the best delay time was obtained when the autocorrelation function drops to $1-1/e$ of its initial value.

Andrew M. Fraser and Harry L. Swinney found a superior criterion in [11], which they called first minimum mutual information, for the choice of time delay. An $O(N \log N)$ algorithm for calculating mutual information is described in their research, and several systems that were applied to this criterion were demonstrated in the paper. N.J.I. Mars in [12] proposed that the average mutual information as the criterion be

applied to non-linear systems. With these methods, an iteration probability density estimation procedure is needed. The frequency histogram method was illustrated by Emanuel Parzen [13] and the kernel density estimator was introduced by Young-II Moon, Balaji Rajagopalan, and UpmanuLall [14].

Besides these traditional methods used to evaluate time delay and embedding dimension, recently Hongguang Ma, Chongzhao Han in [15] illustrated a method for choosing a pair of embedding dimension and time delay operating from the viewpoint that both the embedding dimension and time delay are closely related. The technique is based upon a non-biased multiple autocorrelation approach [16].

After deciding on the embedding dimension and time delay, some researchers began to develop other algorithms to estimate the largest Lyapunov exponent or the spectrum of Lyapunov exponents (excluding the above-mentioned GSR method [6]).

In [17], Ramazan Gencay and W. Davis Dechert gave a detailed introduction to the recursive QR decomposition procedure for finding the eigenvalues of products of matrices and the determination of local Lyapunov exponents for finite time lengths based on QR decomposition. This method is now widely used because it has some advantages over the GSR method: all Lyapunov exponents can be obtained at the same time, and the results are more accurate.

Neural networks were described in many fields of biology before they became rapidly-developing tool used in many fields. This tool can now be used in the calculation of Lyapunov exponents.

Kevin Gurney's book *Introduction to Neural Networks* [18] and Simon Haykin's book *Neural Networks: a Comprehensive Foundation* [19] offered definitions of neural networks, the least mean square algorithm and multilayer feedforward networks.

Ramazan's paper [17] applies the multivariate feed forward network estimation technique to the Lyapunov exponent calculation process. Based on a trained neural network and calculated derivatives of formulated functions, the Lyapunov exponents were evaluated through the QR-decomposition method.

Daniel F. McCaffrey, Stephen Ellner, A. Ronald Gallant and Douglas W. Nychka [20], based on algorithm similar to that in [17], estimate the dominant Lyapunov exponent of a nonlinear dynamic system with additive noise. They also show results from several implementations, and prove that the neural network regression method can provide reliable values for the largest Lyapunov exponent, at least in the case of a Hénon map.

Based on those methods mentioned above, researchers began to use the Lyapunov exponents to study other fields. In Mototsugu Shintani and Oliver Linton's paper [21], Lyapunov exponent estimator using a neural network is used, to develop a statistical framework for testing a chaotic hypothesis and apply it to daily stock return data.

In addition, the papers discussed above are all based on time delay phase space reconstruction, which is commonly used to evaluate Lyapunov exponents. Nevertheless, Pengcheng Xu calculated Lyapunov exponents based on a differential phase space reconstruction instead of time a delay phase space reconstruction, as described in [22].

Theoretically, the chaotic attractor of a differential phase space should be more accurate, and the Lyapunov exponents should be nearly the same to the theoretical values. However, until now, there has been little research in this area.

Besides the above-mentioned evaluation of the largest Lyapunov exponent, other methods to detect the presence or absence of chaos in a dynamical system have been and continue to be developed. The newly developed 0-1 Test method was first derived by Georg A. Gottwald and Ian Melbourne [23]. Their method is applied directly to the time series data and does not require phase space reconstruction. The test developed from statistical methods can be viewed as an algorithm to determine binary quantities. The output of the test can only be 1 or 0, representing chaos or regularity, respectively, for a dynamical system. The authors subsequently published papers [24] and [25] to provide further evidence of the effectiveness of this method.

The above discussion has focused on the time domain of chaotic dynamical systems. Power spectrum analysis provides a new view of the chaotic system analysis: frequency domain. Publications [26] and [27], by Brian D. Storey and H. P. F. Swinnerton-Dyer, respectively, provide different ways to obtain a power spectrum from a scalar time series. Additionally, M C Valsakumar and S V M Satyanarayana [28] investigated the nature of the computed spectral density of a discrete and finite length body of time series data, and performed a comparison between the theoretical power spectrum and the numerical power spectrum. The feature of decay in the high frequency part of the power spectrum in all chaotic dynamical systems could turn out to be a useful

tool for distinguishing the presence or absence of chaos, though this method has not been sufficiently proven.

Though some publications mentioned Lyapunov exponents in some cases for fault detection [29] [30], until now, no one explored the relationship between Lyapunov exponents and fault detection.

1.3 Problem Statement

Fault detection in dynamical systems is a very important issue. However, it is difficult to extract the weak and early fault signal for some nonlinear systems. The largest Lyapunov exponent is usually used to distinguish and to measure chaos of dynamical systems. The exponent or its change can have some relationships with system faults. Usually, the equations of the dynamical systems are difficult to obtain directly; only time series data sets are observable. Thus, researchers must propose methods of calculating the largest Lyapunov exponent from time series data. Based on these methods, the way faults within a dynamical system change the Lyapunov exponents needs to be explored.

1.4 Research Contribution

This research has made the following contributions:

- A method for calculating the Lyapunov exponents from time series data is implemented.
- This method of computing Lyapunov exponents is applied to some simple problems and to a set of experimental data acquired from a real world system to explore the relationship between the largest Lyapunov exponent and incipient fault progression.

1.5 Organization

In Chapter II, a definition of chaos and a description of chaotic dynamical system are both given. To better understand these concepts, some classic examples and well-known attractors are briefly presented. Then the main characteristic of chaos, which is the sensitive dependence on initial conditions, and the Lyapunov exponent, which is the indicator of chaos are both introduced.

In Chapter III, the method of calculating the Lyapunov exponents from a time series data set is illustrated. The first step is that of phase space reconstruction, in which the widely used algorithms for evaluating the embedding dimension (False Nearest Neighbors) and the time delay (Average Mutual Information) are introduced. Then, the QR decomposition and its application to the Lyapunov exponents calculation are derived. Finally, the neural networks, especially the Multilayer Neural Network, is illustrated and applied to the process of estimating the Lyapunov exponents.

In Chapter IV, the complete algorithm developed in Chapter III is used on the data extracted from three different chaotic systems: Lorenz, Hénon and Rössler. In the study of the application to the Lorenz system, the system with perturbed parameters, added noise and inputs are also tested. Furthermore, the effect of changing parameters on a large scale, on the Lyapunov exponents is also exhibited.

In Chapter V, a real world data set acquired from an induction motor is used to study the relationship between fault detection and the Lyapunov exponents. Based on the calculation results obtained by using the method mentioned in Chapter III, the changes of the largest Lyapunov exponent and their distributions through the entire motor bearing damage process are presented and analyzed.

In Chapter VI, the method of calculating the Lyapunov exponent, the performance of the method to several systems and the application of the method on fault detection are summarized and some conclusions are drawn. Rooted in the experimental results, the advantages and drawbacks of the method are demonstrated. In addition, further research possibilities are addressed.

CHAPTER II

INTRODUCTION OF CHAOS

AND LYAPUNOV EXPONENTS

2.1 Introduction

Dynamical systems can be presented by attractors in phase space, and chaotic behavior sometimes occurs in these systems. In this section, a description of chaos is offered and two examples are described: a dripping faucet and a magnet near a superconductor. Then, different varieties of dynamical systems are briefly introduced. The attractors in these systems can be categorized into four groups: fixed point attractors, periodic attractors, quasi-periodic attractors and chaotic attractors. In order to develop a better understanding of these attractors, examples and the phase space appearance for each kind are shown. Furthermore, the main feature of chaos, sensitive dependence on initial conditions, is illustrated. A description of the most important indicator of chaos, the Lyapunov exponent, is offered in the next section. For comparison purposes, other methods of distinguishing chaos (the 0-1 test and the power spectrum method) are briefly discussed. The final section is dedicated to the chapter summary.

2.2 Description of Chaos

The greatest power of science lies in its ability to relate causes to effects. Eclipses, for instance, can be predicted in advance due to the universal law of gravitation. However, there are numerous other natural phenomena that are not predictable, such as weather, the roll of a dice, smoke as it leaves the end of a cigarette, and so on. These phenomena are not easy to predict. These types of phenomena are collectively called chaos. Several researchers have performed experiments geared toward observing and studying chaos and chaotic behavior.

The dripping faucet is an everyday occurrence in our life, but the experiment performed by Shaw (1984) demonstrates how a dripping faucet illustrates chaos. In this experiment, water drops fall from the faucet, and the discrete time intervals between each drop are recorded. When the flow rate is small, the time intervals are equal. As the flow rate is increased to certain level, the time intervals become periodic. When the flow rate is further increased until it is sufficiently high, no apparent regularity of the sequence of time intervals can be observed. This irregularity is an example of chaos. The explanation of this phenomenon is the oscillation of the water at the tip (see in Fig. 2.1). The oscillation has an effect on the initial condition of the following drop. When the flow rate is small, the drops seem to follow some regular pattern. But as the flow rate increases, the variation in initial conditions becomes significant and the result is chaos.



Fig. 2.1. The oscillation of the water at the tip before and after the water drop [31].

Another famous example is Francis Moon's experiment involving a magnet near a superconductor. A sketch of the experiment setup is shown in Fig. 2.2, as taken from [32]. A thin cantilevered beam with a cylindrical rare earth magnet is suspended at the tip. The beam and the magnet act as a pendulum with additional restoring forces due to the bending of the clamped end of the beam. The magnet vibrates above the cylindrical disc of the superconductor. The elastic beam is in the form of a thin steel cantilever and the rare earth cylindrical magnet is attached to the end of the elastic beam with its cylindrical axis transverse to the long direction and parallel to the wide face of the beam. The clamped end of the beam vibrates with a Sinusoidal input motion. A strain gauge, attached to the clamped end of the beam, is used to detect the lateral motion of the magnet. The results obtained from the gauge provide an example of chaotic behavior.

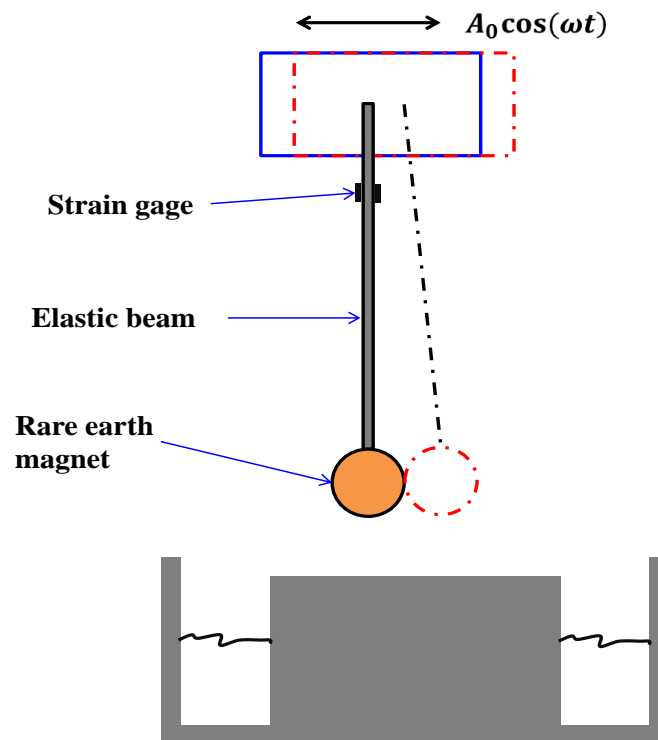


Fig. 2.2. Sketch of a magnet, superconductor, and elastic beam support and excitation apparatus.

2.3 Categories of Dynamical Systems

A dynamical system is defined as a mathematical description for time evolution of a system in a state space. State space is the set of all possible states of a dynamical system, and each state corresponds to a unique trajectory in the space. State space represents the motion of the dynamical system in geometric form. A good example of a dynamical system is a simple pendulum. Its motion is determined by only two variables: position and velocity. Thus, the state of the pendulum is a point in the state space marked by the two coordinates that are the degrees of freedom in the system's motion. The point moves following a path or orbit through the state space as the pendulum swings back and forth in time. If the pendulum is ideal and frictionless, as shown in Fig.

2.3 (left), the orbit is a closed curve through a sequence of the states, and the attractor (an attractor is the behavior into which the system settles, or to which it is attracted) is called a limit cycle. If the pendulum experiences friction, as shown in Fig. 2.3 (right), the orbit spirals to a fixed point because the pendulum eventually comes to a halt. This orbit is known as a fixed point attractor.

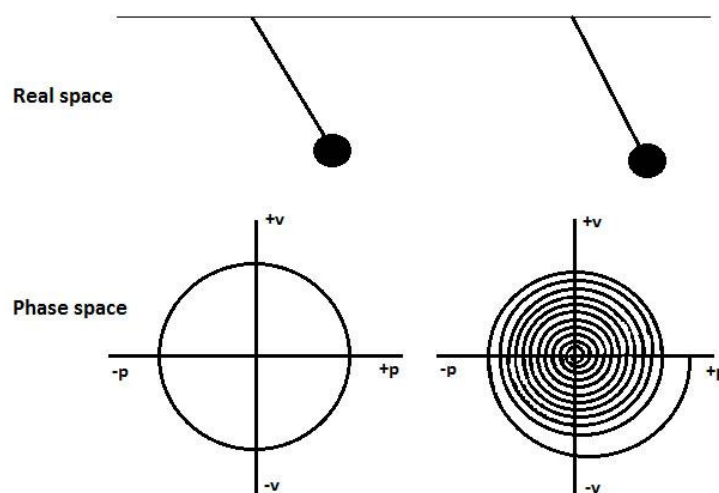


Fig. 2.3. The real space and state space of a pendulum with and without friction.

The next important and more complicated attractor is a torus, which is the surface of revolution generated by revolving a circle within a three-dimensional space about an axis coplanar with that circle. It looks like a doughnut, as shown in Fig. 2.4. Because two independent oscillations make up this shape of motion, it is also called quasi-periodic motion. Furthermore, some attractors can be of higher dimensional torus when they represent a combination of more than two oscillations.

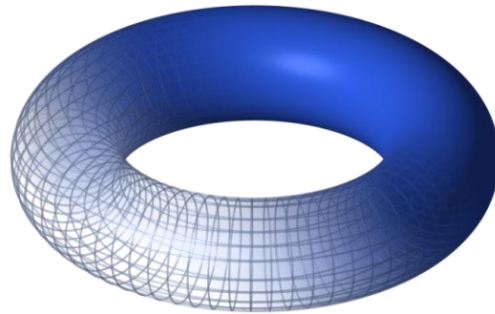


Fig. 2.4. The appearance of a torus attractor [33].

Although a torus is much more complicated than a fixed point attractor and a limit cycle, quasi-periodic motion is still predictable. Compared with the three above-mentioned predictable systems, chaotic dynamical systems have unpredictable behavior. In the state space, the chaotic behavior stems from the repeated operation of stretching and folding, so the chaotic attractor (which is also called the strange attractor) has a much more complicated structure than the other three attractors. The chaotic attractor does not have a smooth surface, but instead has folds at large scales. Fig. 2.5 shows the chaotic attractor of a Van der Pol equation, which is one of the simplest chaotic systems.

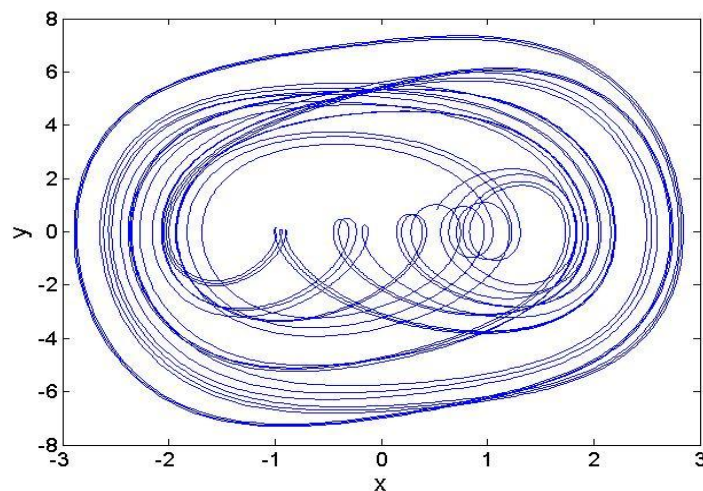


Fig. 2.5. The phase space of a Van der Pol attractor.

The evolution of a dynamical system may occur either in continuous time or in discrete time. The former is called flow, and the latter is called map. For nonlinear systems, continuous flow and discrete map are also two mathematical concepts used to model chaotic behavior.

The earliest and the most famous concrete example of low dimension continuous flow for chaotic dynamics is the Lorenz system, which was discovered by Edward N. Lorenz of the Massachusetts Institute of Technology in 1963. Motivated by a desire to understand the unpredictability of the weather, he simplified the motion equations to three dimensions and obtained an attractor now known as the Lorenz attractor.

A frequently found example in the literature of a discrete map is the Hénon map, which is a model invented to investigate the properties of chaos. Detailed descriptions of the Lorenz system and the Hénon map are provided in Chapter IV.

2.4 Sensitive Dependence upon Initial Conditions

For a simple dynamical system such as a frictionless pendulum, motion is completely determined by the initial conditions. The three kinds of predictable attractors mentioned above all have the same feature: if the starting points of the orbits are near to one another, they will remain near to one another at all future times.

However, chaotic dynamical systems exhibit behavior that is extremely sensitive or exponentially sensitive and dependent upon initial conditions. The stretching and folding operations gradually remove the initial information and replace it with other information. After a short time interval, all causal connections between the past and the

future are lost. Thus, small variations in initial conditions produce differences that vary exponentially over time. This is the main property of chaotic systems that make them different from other systems.

In fact, exponential divergence is a local feature because attractors have a finite size and the two orbits that form a chaotic attractor cannot diverge forever. The orbits follow different paths, but eventually pass close to one another due to the stretching and folding.

For non-chaotic dynamical systems, since the nearby points stay close in the future, the measured errors are not enlarged. In other words, the non-chaotic systems are not sensitive to measurement errors. For chaotic dynamical systems, the stretching operation may make small-scale uncertainties larger and the folding operation may bring widely separated trajectories together and erase large scale bodies of information. In this light, a tiny change could have a huge effect on a chaotic system. A frequently mentioned poetic description of chaos is that a butterfly in China can cause a hurricane in the Atlantic.

2.5 Indicators of Chaos

Chaos can be observed in a time series or in a phase space plot, but this is not very accurate. Therefore, the problem of detecting and qualifying chaos has become an important issue for researchers. At present, there are a number of methods employed to detect chaos in dynamical systems, such as power spectrum analysis, the 0-1 test, calculating the Lyapunov exponents, and so on.

A power spectrum is a plot of a given signal's power or energy per unit time through different frequencies. The most common way to generate a power spectrum is based on a Fourier transform and autocorrelation. Power spectrum analysis provides a framework for analyzing chaos. For chaotic systems, a power spectrum has features of broadband signals and its tail is exponentially decaying at high frequencies. These features can be used to detect the existence of chaos.

A recently developed method is the 0-1 Test which is used to extract a binary quantity from a power spectrum. It is designed to distinguish between regular and chaotic dynamics for a deterministic dynamical system. The inputs are time series data and the output is 1, which represents chaos or 0 which represents non-chaos.

The Lyapunov exponents, especially the largest Lyapunov exponent, of a dynamical system have been shown to be the most important and most useful of a number of invariants, for fundamentally characterizing attractors and chaos. This exponent not only shows whether the system experiences chaos, but it also gives the level of the chaos within the system.

In mathematics, the Lyapunov exponent of a dynamical system is a quantity that characterizes the rate of separation of infinitesimally close trajectories. If the initial separation of two trajectories in phase space can be expressed as a vector and denoted as $\delta\mathbf{X}(0)$, the divergence after some time t can be represented as

$$\|\delta\mathbf{X}(t)\| = e^{\lambda t} \|\delta\mathbf{X}(0)\| \quad (2.1)$$

where $\delta\mathbf{X}(t)$ is the separation vector of the two state vectors of the two trajectories, and the exponent λ varies with time. The properly averaged exponent, λ , is called the

Lyapunov exponent. For systems with more than one dimension, the rate of separation can be different for the different orientations of the initial separation vector.

Thus the Lyapunov exponent is precisely defined as the average exponential rate of divergence or convergence of two trajectories with nearby initial conditions. In a dynamical system or phase space with d dimensions, the separation rate of different orientations may be different and the number of Lyapunov exponents should be equal to the number of dimensions d . This group of Lyapunov exponents ordered from the largest to the smallest is called the spectrum of Lyapunov exponents, and the largest exponent which called the Maximal or Largest Lyapunov Exponent (MLE or LLE), is usually considered to be the indicator of chaos because it determines the predictability of a dynamical system. A positive Lyapunov exponent denotes a system to be chaotic, and a negative one indicates no chaos.

The expression used to calculate the Lyapunov exponents can be stated as

$$\lambda_i = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \log \frac{\sigma x_i(n+1)}{\sigma x_i(n)}, \quad (2.2)$$

for a discrete system; and

$$\lambda_i = \lim_{\sigma x_i(0) \rightarrow 0} \lim_{t \rightarrow \infty} \frac{1}{t} \log \frac{\sigma x_i(t)}{\sigma x_i(0)}, \quad (2.3)$$

for a continuous system. $x_i(n)$ or $x_i(t)$ is the i th state of the state vector. When N or t is finite, λ is also called the local Lyapunov exponent.

The LLE is a useful tool to distinguish various types of orbits and the characteristics of different dynamical systems. When the LLE is negative, the orbit attracts a stable fixed point or a stable limit cycle. A negative LLE is a characteristic of a

dissipative or non-conservative system. Such systems exhibit asymptotic stability; the more negative the exponent, the greater the stability. When the LLE is zero, the orbit is a neutral fixed point (or an eventually fixed point). An LLE of zero indicates that the system is in some sort of a steady state mode. A physical system with zero exponent is conservative. Such systems exhibit Lyapunov stability. When the LLE is positive, then the orbit is unstable and chaotic. Nearby points, no matter how close they are, will diverge to any arbitrary separation. All neighborhoods in the phase space will eventually be visited. For a discrete system, the orbits will look like snow on a television set. For a continuous system, the phase space will be a tangled sea of wavy lines, like in a pot of spaghetti.

2.6 Chapter Summary

In this chapter some basic concepts of chaos were introduced, along with some famous examples from the literature. Then, the demonstration of different kinds of dynamical systems and their attractor appearances in phase space led to an unpredictable dynamical system called a chaotic dynamical system. For chaos, sensitivity to initial conditions is the most distinguishable character trait, and its measurement is used in determining the Lyapunov exponent. The definition equation for calculating the Lyapunov exponent was stated in this chapter, but in reality the equations and the states of systems are not always available. Thus, calculating the Lyapunov exponents from time series data becomes an important problem to solve. This issue is presented in the next chapter.

CHAPTER III

COMPUTING THE LYAPUNOV EXPONENTS

FROM MEASUREMENTS

3.1 Introduction

Based on the information presented in Chapter II, it is clear that calculation of the LLE is significant for an accurate analysis of chaos in dynamical systems. This chapter focuses on the methods for calculating the LLE. In the next section, a method based on the previously discussed definition of the Lyapunov exponent is briefly introduced. Then, Wolf's algorithm to evaluate the LLE directly from a time series data set is discussed. An implementation of this algorithm is also illustrated. And then a more comprehensive method is described in the following sections, in an effort to achieve more accurate results. The first step in this method is phase space reconstruction, the key points of which are the decisions regarding time delay and the embedding dimension. The next step is to attain the sub-functions and their derivatives by using neural networks. The last step is to formulate the Jacobi matrices, and calculate the Lyapunov exponents based on those matrices by using the QR decomposition.

3.2 Problems with Computing the Lyapunov Exponents

If all of the relevant information in the system is well known, the calculation of the theoretical Lyapunov exponents can be based on the equations of that system. This method includes repeatedly using equation linearization and the Gram-Schmidt Reorthonormalization (GSR) procedure on the vector frame. The Lyapunov exponents are calculated, then, from the growth rates of the vectors.

In reality, the equations in a given system are not easy to obtain. However, time series data sets can easily be acquired. When only time series data are recorded, the calculation method introduced above is impossible to use. Therefore, the problem of computing the Lyapunov exponents directly from a time series is significant to researchers in this area.

Alan Wolf [6] offers an algorithm to use in order to compute the LLE from time series data. The general idea is to follow two nearby orbits and calculate their average logarithmic rate of separation. Whenever they get too far apart, one of the orbits must be moved back to the vicinity of the other along the line of separation. Based on this method, J. C. Sprott published a conservative procedure for calculating the LLE [34]. This procedure can briefly be summarized as follows:

Step 1: Start with any initial condition in the basin of attraction. A better choice would be to start with a point known on the attractor, in which case *Step 2* can be omitted.

Step 2: Iterate until the orbit is on the attractor. This requires some judgment or prior knowledge of the system under study. For most systems, it is safe only to iterate a few hundred times and assume that is sufficient.

Step 3: Select a nearby point with a separation of d_0 . An appropriate choice of d_0 is one that is about 1000 times larger than the precision of the floating point numbers that are being used. For example, in a case of (8-byte) double-precision (the minimum recommended for such calculations), variables have a 52-bit mantissa, and the precision is calculated as $2^{-52} = 2.22 \times 10^{-16}$. Therefore, a value of $d_0 = 10^{-12}$ will usually be sufficient.

Step 4: Advance the two orbits one iteration and calculate the new separation d_1 . The new separation is the distance between the two new points in the phase space. So, for a 2-dimensional system with variables x and y , the separation would be $d = [(x_a - x_b)^2 + (y_a - y_b)^2]^{1/2}$, where the subscripts (a and b) denote the two orbits, respectively.

Step 5: Evaluate $\log|d_1/d_0|$ in any convenient base. By convention, the natural logarithm (base-e) is usually used, but for maps, the Lyapunov exponents are often quoted in bits per iteration, in which case one would need to use base-2. (Note that $\log_2 x = 1.4427 \log_e x$).

Step 6: Readjust one orbit so its separation is d_0 in the same direction as d_1 . This is probably the most difficult and error-prone step. For example (in 2-dimensions), suppose orbit b is to be adjusted and its value after one iteration is (x_{b1}, y_{b1}) . It would then be reinitialized to $x_{b0} = x_{a1} + d_0(x_{b1} - x_{a1})/d_1$ and $y_{b0} = y_{a1} + d_0(y_{b1} - y_{a1})/d_1$.

Step 7: Repeat Steps 4-6 many times and calculate the average of Step 5. It is better to discard the first few values obtained to be sure the orbits have oriented themselves along the direction of maximum expansion. If the system is a continuous flow that consists of

ordinary differential equations, the procedure is the same except that the result of the exponent should be derived by the iteration step size. In such a case, the unit of the exponent is inverse seconds instead of inverse iterations.

However, the method is only good for estimating the first few non-negative Lyapunov exponents, and it offers better results for discrete rather than continuous systems. Thus, a method which can be more robust and more widely used needs to be developed. Fig. 3.1 shows a flow chart of the method used in this research, and the details of each part are introduced in Sections 3.3 to 3.5.

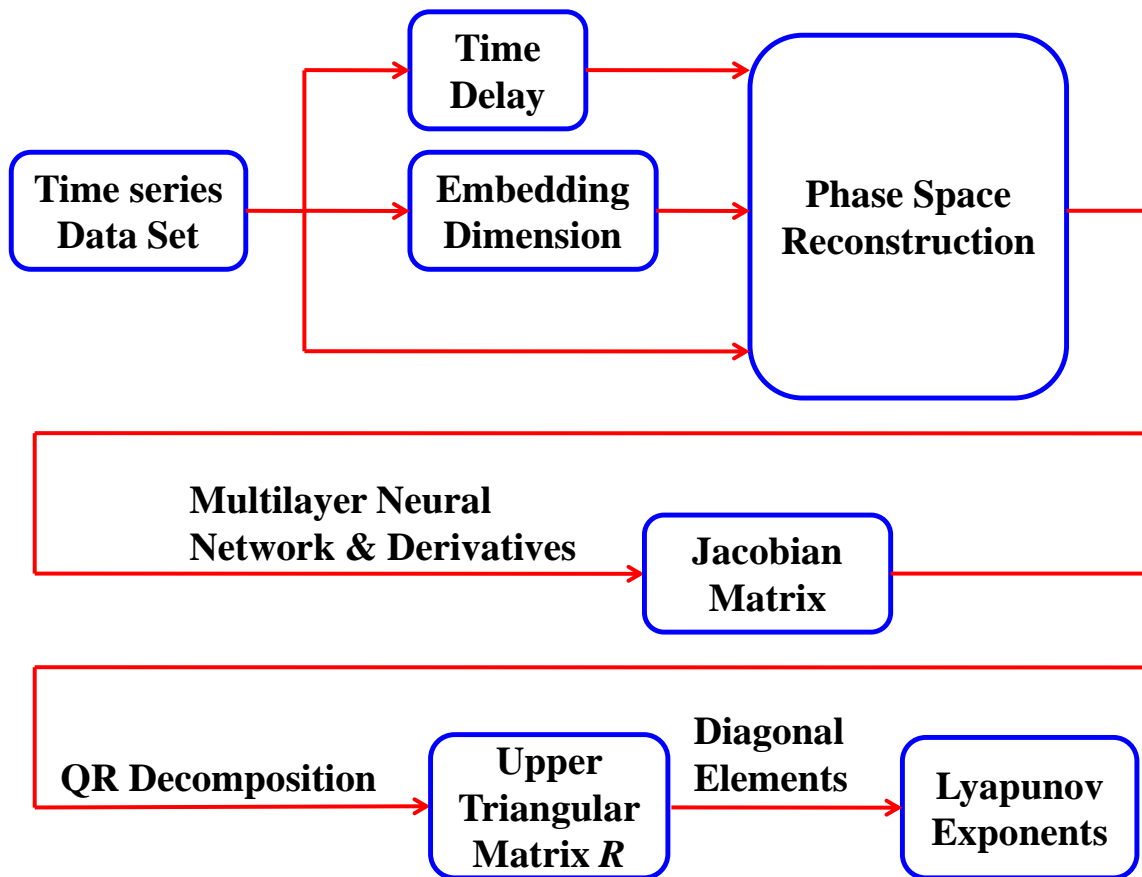


Fig. 3.1. Flow chart for calculating Lyapunov exponents.

3.3 Phase Space Reconstruction

In a dynamical system, phase space or state space is a collection of possible coordinates needed to describe the system. However, for an experiment, the available information is usually not a phase space but only time series data from some of the states. Therefore the problem of converting time series data into an induced state space is well known, and commonly referred to as phase space reconstruction. For phase space reconstruction, the time-delay method is the most useful technical solution, and is summarized below.

Assume a discrete time series data set:

$$\{x(0), x(1), \dots, x(n), \dots, x(N-1), x(N)\}.$$

From the data set, a general time-delay vector can be formed as:

$$\mathbf{y}_d(n) = [x(n), x(n + \tau), \dots, x(n + (d - 1)\tau)]^T; \quad (3.1)$$

$$n = 0, 1, 2, \dots, N - (d - 1)\tau,$$

where “T” denotes “transpose”, τ is a suitable time delay and d is the embedding dimension. For a nonlinear dynamical system, when the embedding dimension is large enough, the information regarding the behavior of the system can be fully represented regardless of which variable of the system is measured. The time delay τ should not be too large, because too large a τ can cause two nearby vectors to be independent and lose their relationship to one another.

Thus, there are two main quantities that need to be decided upon to reconstruct the phase space from the measured time series data of a given dynamical system: the

time delay and the embedding dimension. A number of methods exist for evaluating a suitable time delay and embedding dimension.

On the one hand, there are several functions that can be used to evaluate the time delay: autocorrelation and power spectrum functions, Average Mutual Information functions (AMI); degree of separation functions. However, the AMI method is already well developed and commonly used. On the other hand, analytical methods used to determine the embedding dimension include: the False Nearest Neighbors (FNN) method, the Bad Prediction method (BP), and application of the fractal and the correlation dimensions. The FNN method is the most widely used. An introduction of the Average Mutual Information and the False Nearest Neighbors methods are provided in Sections 3.3.1 and 3.3.2, respectively.

3.3.1 Average Mutual Information

According to probability theory, mutual information is a measurement of the mutual dependence of two random variables, and this quantity can reflect the similarity of those two variables. The data collected through experiments or via a computer is always a sampled type of data, so mutual information is only considered in the discrete type of situation described. Two groups of finite numbers of values are assumed as $X = \{x_i\}, i=1, \dots, N$ and $Y = \{y_j\}, j=1, \dots, M$, which are used to develop the expression of the mutual information.

The occurrence probability of x_i in the group X and the occurrence probability of y_j in group Y are respectively denoted as $p_X(x_i)$ and $p_Y(y_j)$. The joint occurrence probability is defined as:

$$p_{XY}(x_i, y_j) = p(Y = y_j | X = x_i) p(X = x_i) = p(X = x_i | Y = y_j) p(Y = y_j). \quad (3.2)$$

Therefore,

$$\sum_{i=1}^N \sum_{j=1}^M p_{XY}(x_i, y_j) = 1. \quad (3.3)$$

In an experiment, any of these probabilities can be estimated by certain methods such as frequency histogram or kernel density estimation.

The mutual information between these two groups of data can be defined as

$$I(X, Y) = \log \frac{p_{XY}(x_i, y_j)}{p_X(x_i) p_Y(y_j)}. \quad (3.4)$$

If X and Y are independent, their joint probability $p_{XY}(x_i, y_j)$ is equal to the product of $p_X(x_i)$ and $p_Y(y_j)$, so that the mutual information is zero.

Furthermore, the average mutual information of these two groups of data is defined as:

$$AMI(X, Y) = \sum_{i=1}^N \sum_{j=1}^M p_{XY}(x_i, y_j) \log \frac{p_{XY}(x_i, y_j)}{p_X(x_i) p_Y(y_j)}. \quad (3.5)$$

This is useful for identifying sampled multivariate as either related or independent. The original time series data set is considered to be X , and the time delay time series data set is taken as Y . The first minimum of the average mutual information is usually proposed as a criterion for choosing the suitable time delay τ . This choice is better than other criteria such as that obtained from the autocorrelation function, especially for the situation of nonlinear systems.

3.3.2 False Nearest Neighbors Method

The False Nearest Neighbors (FNN) method is a way of using previous vectors to predict future vectors based only on the properties of the data set itself. In a scalar of well sampled data, the embedding dimension of the reconstructed phase space is chosen as follows: if the embedding dimension is very small, then points near the space coordinates may be due to the folding effect of the projection and not the close data in the original dynamical system. If this is the case, the predictions from this phase space are not convincing. These close points caused only by folding are known as false neighbors. To detect false neighbors, vectors from the data are constructed in dimensions 1, 2, 3 and so forth, and the fractions of the nearest neighbors in the data set are calculated. After increasing the dimension several times until the dimension d for which all the nearest neighbors are also the nearest neighbors in dimension 1, the necessary minimum embedding dimension can be found.

The steps of the False Nearest Neighbors method can be summarized as follows:

Step 1: Find the closest point and its distance to a given point in the time delay reconstructed phase space. For instance, the vector $\mathbf{y}_d(k) = (x(k), x(k + \tau), \dots, x(k + (d - 1)\tau))$ can be also viewed as a point in a d dimensional phase space. After searching all the points in this space, the point $\mathbf{y}_d(j) = (x(j), x(j + \tau), \dots, x(j + (d - 1)\tau))$ which is closest to $\mathbf{y}_d(k)$ in the Euclidean sense can be found. $\mathbf{y}_d(j)$ is known as the nearest neighbor to $\mathbf{y}_d(k)$.

Step 2: Distinguish the false neighbors and the real neighbors, using the flow chart shown in Fig. 3.2. The fraction of nearest neighbors is calculated and compared with a

previously given threshold value R . If the fraction is smaller than R , the neighbors are real neighbors. Otherwise, the neighbors are considered false neighbors.

Step 3: Apply the *Step 1* and *Step 2* to all data in the data set and compute the percentage of the false nearest neighbors from all data.

Step 4: Increase the dimension d until the percentage frequency of the false nearest neighbors is at an acceptably low level such that this d can be considered an ideal embedding dimension.

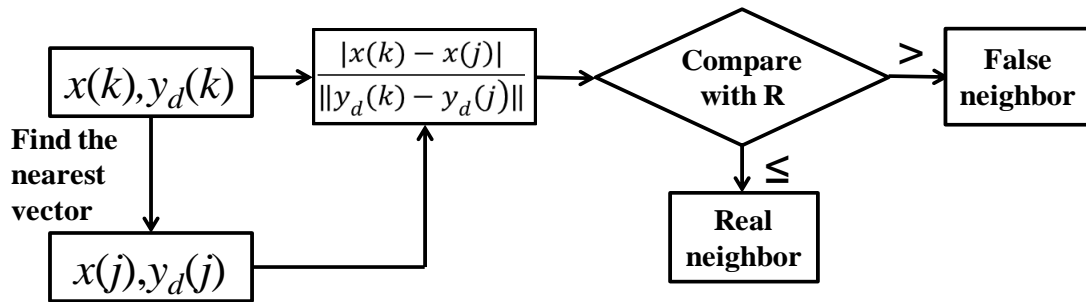


Fig. 3.2. Flow chart for distinguishing false neighbors and real neighbors.

After determining the time delay and the embedding dimension, the phase space can be reconstructed from the data set by using Equation (3.1).

3.4 Jacobian Matrix and the QR Decomposition

The definition and the theoretical calculation of Lyapunov exponents were introduced in previous sections. Wolf's method [6] for calculating Lyapunov exponents from time series data has also been summarized. In this section, a method for evaluating the full spectrum of Lyapunov exponents by calculating the eigenvalues of a Jacobian matrix is stated.

If a time series of data X is available for a dynamical system, a phase space for the system can be reconstructed, as discussed in Section 3.3. The time delay vector can be stated as $\mathbf{y}_d(n) = [x(n), x(n + \tau), \dots, x(n + (d - 1)\tau)]^T$, where τ is the time delay and d is the dimension of the vector $\mathbf{y}_d(n)$. In the time series of data X with N samples, there should be $(N-d+1)$ time delay vectors and $(N-d)$ transfer maps (functions) $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_n, \dots, \mathbf{g}_{N-d-1}$ where \mathbf{g}_n is the map from one vector $\mathbf{y}_d(n)$ to the next vector $\mathbf{y}_d(n + 1)$. The relationship between the two adjacent time delay vectors, such as $\mathbf{y}_d(n)$ and $\mathbf{y}_d(n + 1)$, can be stated as follows:

Assume the value of data $x(n + d\tau)$ can be calculated from the previous n data $x(n), x(n + \tau), \dots, x(n + (d - 1)\tau)$ by a function f_n .

$$x(n + d\tau) = f_n(x(n), x(n + \tau), \dots, x(n + (d - 1)\tau)), \quad (3.6)$$

Then,

$$\mathbf{g}_n : \mathbf{y}_d(n) \rightarrow \mathbf{y}_d(n + \tau)$$

$$\mathbf{g}_n : \begin{bmatrix} x(n) \\ x(n + \tau) \\ \vdots \\ x(n + (d - 2)\tau) \\ x(n + (d - 1)\tau) \end{bmatrix} \rightarrow \begin{bmatrix} x(n + \tau) \\ x(n + 2\tau) \\ \vdots \\ x(n + (d - 1)\tau) \\ x(n + d\tau) \end{bmatrix} = \begin{bmatrix} x(n + \tau) \\ x(n + 2\tau) \\ \vdots \\ x(n + (d - 1)\tau) \\ f_n(x(n), x(n + \tau), \dots, x(n + (d - 1)\tau)) \end{bmatrix}, \quad (3.7)$$

The derivative of the map \mathbf{g}_n is

$$\mathbf{Dg}_n = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ & & & & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ \frac{\partial f_n}{\partial x(n)} & \frac{\partial f_n}{\partial x(n+\tau)} & \cdots & \cdots & \cdots & \frac{\partial f_n}{\partial x(n+(d-1)\tau)} \end{bmatrix}, \quad (3.8)$$

which is also called the Jacobian matrix.

For a dynamical system with d states, there should be d Lyapunov exponents, represented as $\{\lambda_1, \lambda_2, \dots, \lambda_d\}$, and ordered from the largest to the smallest where λ_1 is the largest.

The product of $\mathbf{Dg}_{N-d-1}, \mathbf{Dg}_{N-d-2}, \dots, \mathbf{Dg}_1, \mathbf{Dg}_0$ is renamed as $\mathbf{J}^{(N-d)\tau}(\mathbf{y}_d(0))$. $\mathbf{y}_d(0)$ is the initial time delay vector, and can be expressed as the transpose of $[x(0) \ x(\tau) \ \dots \ x((d-1)\tau)]$. The spectrum of Lyapunov exponents with this initial vector can be represented as

$$\lambda(\mathbf{X}_0) = \begin{bmatrix} \lambda_1(\mathbf{y}_d(0)) \\ \lambda_2(\mathbf{y}_d(0)) \\ \vdots \\ \lambda_d(\mathbf{y}_d(0)) \end{bmatrix}, \quad (3.9)$$

and the i th Lyapunov exponent $\lambda_i(\mathbf{y}_d(0))$ can be evaluated as

$$\lambda_i(\mathbf{y}_d(0)) = \log \mathbf{A}_i(\mathbf{y}_d(0)), \quad (3.10)$$

where $\mathbf{A}_i(\mathbf{y}_d(0))$ is the i th eigenvalue of $\mathbf{L}(\mathbf{y}_d(0))$, and where

$$\mathbf{L}(\mathbf{y}_d(0)) = \lim_{N \rightarrow \infty} (\mathbf{J}^{(N-d)\tau}(\mathbf{y}_d(0)) \cdot [\mathbf{J}^{(N-d)\tau}(\mathbf{y}_d(0))]^T)^{1/(2(N-d)\tau)}. \quad (3.11)$$

All of the Lyapunov exponents can be evaluated by calculating the Jacobian matrices of all the maps. There is a convenient and commonly used method to extract the eigenvalues called the QR decomposition, which can significantly simplify the calculation process.

In linear algebra, QR decomposition (also called QR factorization) of a matrix M states that any matrix can be decomposed into an orthogonal matrix and an upper triangular matrix, so that

$$M = Q \cdot R, \quad (3.12)$$

where Q is the orthogonal matrix, and R is the upper triangular matrix. If $Dg_n Q_{n-1}$ is considered as the matrix M and the QR decomposition is applied to it, then

$$Dg_n Q_{n-1} = Q_n R_n. \quad (3.13)$$

Applying the QR decomposition to the product $Dg_n Q_{n-1}$ aims to build the relationship between Q_{n-1} and Q_n . Based on this equation, $Dg_0 Q_{-1} = Q_0 R_0$, in which the first matrix Q_{-1} is initialized to be an identity matrix. Then the following formulation is obtained:

$$\begin{aligned} Dg_n, \dots, Dg_1 Dg_0 &= (Q_n R_n Q_{n-1}^{-1})(Q_{n-1} R_{n-1} Q_{n-2}^{-1}), \\ \dots, (Q_1 R_1 Q_0^{-1})(Q_0 R_0 Q_{-1}^{-1}) &= Q_n R_n R_{n-1}, \dots, R_1 R_0. \end{aligned} \quad (3.14)$$

From Equation (3.14), it is simple to find the eigenvalues of the R matrices, because they are upper triangular matrices and their eigenvalues are simply their diagonal elements. The eigenvalues of the products of all the R matrices are the products of the corresponding diagonal elements. In addition, if the number of the sample N is

large enough, then the matrix \mathbf{Q}_n can be ignored. Based on Equations (3.10) (3.11) and (3.14), the full Lyapunov exponents can be stated as:

$$\begin{aligned}
\lambda(y_d(0)) &= \log \Lambda(y_d(0)) \\
&= \log(\text{eigenvalues}(L(y_d(0))) \\
&= \log(\text{eigenvalues}((\mathbf{J}^{(N-d)\tau}(y_d(0)) \bullet [\mathbf{J}^{(N-d)\tau}(y_d(0))]^T)^{\frac{1}{2(N-d)\tau}})) \\
&= \frac{1}{2(N-d)\tau} \log(\text{eigenvalues}(\mathbf{J}^{(N-d)\tau}(y_d(0))[\mathbf{J}^{(N-d)\tau}(y_d(0))]^T)) \\
&= \frac{1}{2(N-d)\tau} \log(\text{eigenvalues}(\mathbf{Dg}_{N-d-1}, \dots, \mathbf{Dg}_1 \mathbf{Dg}_0) \\
&\quad \bullet \text{eigenvalues}([\mathbf{Dg}_{N-d-1}, \dots, \mathbf{Dg}_1 \mathbf{Dg}_0]^T)) \\
&= \frac{1}{2(N-d)\tau} \log(\text{eigenvalues}(\mathbf{Q}_{N-d-1} \mathbf{R}_{N-d-2}, \dots, \mathbf{R}_1 \mathbf{R}_0) \\
&\quad \bullet \text{eigenvalues}([\mathbf{Q}_{N-d-1} \mathbf{R}_{N-d-1} \mathbf{R}_{N-d-2}, \dots, \mathbf{R}_1 \mathbf{R}_0]^T)) \\
&= \frac{1}{2(N-d)\tau} \log(\text{eigenvalues}(\mathbf{R}_{N-d-2}, \dots, \mathbf{R}_1 \mathbf{R}_0) \\
&\quad \bullet \text{eigenvalues}([\mathbf{R}_{N-d-1} \mathbf{R}_{N-d-2}, \dots, \mathbf{R}_1 \mathbf{R}_0]^T))
\end{aligned} \tag{3.15}$$

and

$$\begin{aligned}
\lambda_i(X_0) &= \frac{1}{2(N-d)\tau} \log \left((R_{N-d-1}^{ii} R_{n-d-2}^{ii}, \dots, R_1^{ii} R_0^{ii})^2 \right) \\
&= \frac{1}{(N-d)\tau} \sum_{n=0}^{N-d-1} \log[R_n^{ii}],
\end{aligned} \tag{3.16}$$

where R_n^{ii} represents the i th element of the diagonal of the matrix \mathbf{R}_n .

The requirement of the QR decomposition is the acquisition of the transfer maps of connecting the time delay vectors and then the derivatives. The implementation of acquiring these maps is best realized by using neural networks. The concept and the implementation will be introduced in the next section.

3.5 Artificial Neural Networks

3.5.1 Introduction of Artificial Neural Networks

The term “artificial neural networks” refers to a computational paradigm that can be said to mimic the processing within the human brain. Many researchers have offered a variety of explanations of this term. One relatively accurate definition that encompasses most of the key concepts for this study is as follows: “an artificial neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by the process of adaptation to, or learning from, a set of training patterns” [18].

In artificial neural networks, the basic unit is the artificial neuron, whose structure is shown in Fig. 3.3. The inputs x_0, x_1, \dots, x_p are multiplied by weights $w_{k0}, w_{k1}, \dots, w_{kp}$ respectively, and the weighted values are fed to the summing junction. The sum is added by a bias b to form the net input v_k which is the argument of the activation function. The activation is then compared to a threshold; if the activation exceeds the threshold, the unit produces a high valued output. Otherwise, its output is zero.

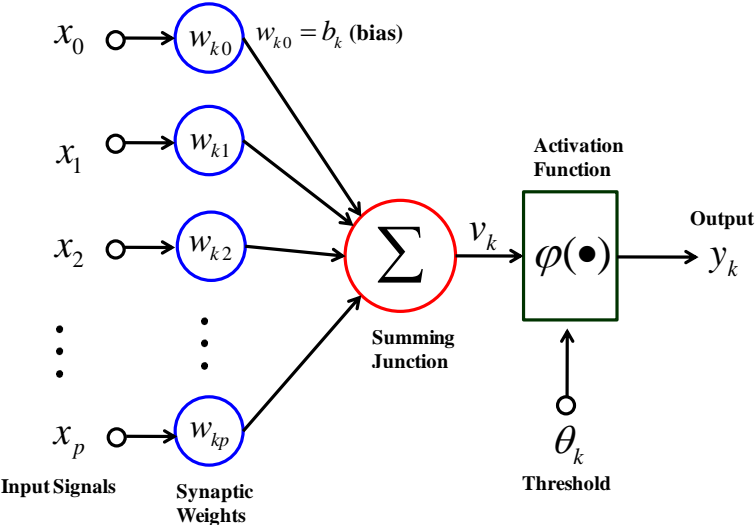


Fig. 3.3. The structure of an artificial neuron.

One neuron forms the simplest network. More neurons can be combined in a layer, and networks can contain several such layers. One example of a multilayer neural network is shown in Fig. 3.4. Each neuron is represented by a circle.

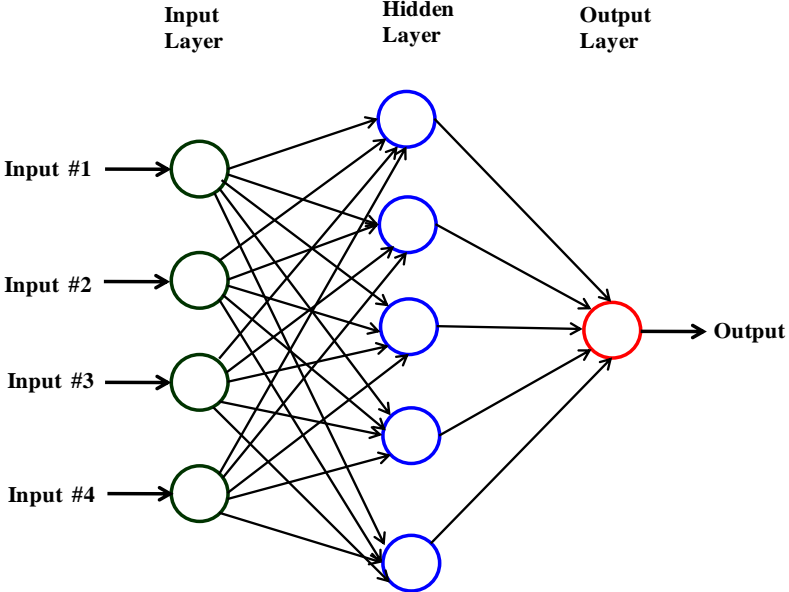


Fig. 3.4. The structure of a multilayer neural network.

3.5.2 Neural Network Learning

Suppose that a set of input vectors and a set of target vectors which are both given. Based on the feedforward neural network and the inputs, the outputs are obtained corresponding to the target set. For each input vector, the difference or error between an output vector and a target vector can be calculated. In order to minimize this error, the network is trained according to the “Adaline” or “Least Mean Square” (Widrow-Hoff) adaptation rule.

The Least Mean Square (LMS) algorithm is also known as the delta rule or the Widrow-Hoff rule; it operates with a single linear neuron model. As with the scheme shown in Fig. 3.5, x_0, x_1, \dots, x_p is a group of input data; w_0, w_1, \dots, w_p is the set of weights. The activation function is linear. The aim is to determine the optimum set of weights in order to minimize the difference between the output of the system and the desired value, in the sense of a mean-square.

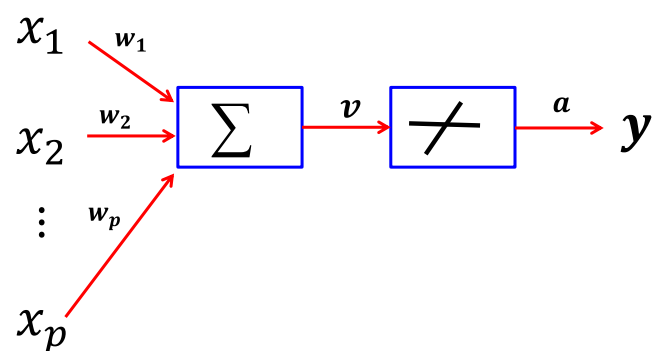


Fig. 3.5. A single neuron structure.

Before we can have a proper discussion of the LMS algorithm, Wiener-Hoff equations need to be introduced. Reference [19] provides the detailed derivation of the equations.

The input-output equation can be simply written as:

$$y = \sum_{k=1}^p w_k x_k . \quad (3.17)$$

The difference between the desired value and the output is defined as:

$$e = d - y , \quad (3.18)$$

where d is the target value.

The mean-squared error is defined as:

$$J = \frac{1}{2} \sum e^2 . \quad (3.19)$$

After substituting Equations (3.17) and (3.18) into equation (3.19) and interchanging the order of expectation and summation due to the linear property, the new equation obtained is

$$J = \frac{1}{2} d^2 - \sum_{k=1}^p [w_k \sum (x_k d)] + \frac{1}{2} \sum_{j=1}^p \sum_{k=1}^p [w_j w_k \sum (x_j x_k)] \quad (3.20)$$

The equation is simplified as:

$$J = \frac{1}{2} d^2 - \sum_{k=1}^p w_k r_{dx}(k) + \frac{1}{2} \sum_{j=1}^p \sum_{k=1}^p w_j w_k r_x(j, k) \quad (3.21)$$

where $r_{dx}(k) = \sum (x_k d)$, $k=1, 2, \dots, p$ is the cross-correlation function between the target value and the input data x_k ; and $r_x(j, k) = \sum (x_j x_k)$, $j, k=1, 2, \dots, p$ is the autocorrelation function of the set of inputs.

To obtain the minimum value of J , the cost function J is differentiated with respect to the weight w_k , and the result is set at zero for all k . The derivative of J with respect to w_k is called the gradient, and is represented by,

$$\nabla_{w_k} J = \frac{\partial J}{\partial w_k}, \quad k=1, 2, \dots, p. \quad (3.22)$$

Finally, the Wiener-Hoff equations are obtained as follows:

$$\sum_{j=1}^p w_{0j} r_x(j, k) = r_{xd}(k), \quad k=1, 2, \dots, p. \quad (3.23)$$

The LMS algorithm is based on the use of instantaneous estimated values for the cost function J . The inputs x_0, x_1, \dots, x_p and the weights w_0, w_1, \dots, w_p are viewed as two vectors $\mathbf{x}=[x_0, x_1, \dots, x_p]$ and $\mathbf{w}=[w_0, w_1, \dots, w_p]$.

From Equation (3.17) and (3.18), the error for the n th iteration is expressed below:

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n), \quad (3.24)$$

Thus,

$$\frac{\partial e(n)}{\partial w(n)} = -\mathbf{x}(n), \quad (3.25)$$

and,

$$\frac{\partial J(n)}{\partial w(n)} = e(n) \frac{\partial e(n)}{\partial w(n)} = -\mathbf{x}(n)e(n). \quad (3.26)$$

Based on established rules of calculus, the partial derivative of the error of the network or the mean square error with respect to each weight w_k can help us learn the direction in which the network error is moving. The negative of this derivative is taken and added to the weight in order to decrease the error until a local minimum value is reached. This process can be described through an iteration equation as follows:

$$\mathbf{w}(n + 1) = \mathbf{w}(n) - \eta \frac{\partial J(n)}{\partial \mathbf{w}(n)}, \quad (3.27)$$

where η is the learning rate parameter.

This algorithm is also known as the gradient descent or method of steepest descent.

Substituting Equation (3.26) into equation (3.27), the formulation of the LMS algorithm can be stated as follows:

$$\hat{\mathbf{w}}(n + 1) = \hat{\mathbf{w}}(n) + \eta \mathbf{x}(n)e(n), \quad (3.28)$$

where $\hat{\mathbf{w}}$ instead of \mathbf{w} emphasizes that the weight vector is an estimate.

3.5.3 Multilayer Feedforward Networks

A multilayer feedforward network is a kind of network which consists of at least three layers (the input layer, the hidden layer and the output layer), with each layer downstream of another layer acting as the input vector. Usually, multilayer feedforward networks consist of three layers, but in some special cases four layers (two hidden layers) are required.

There are three characteristics of multilayer feedforward networks: (1) the network must include a nonlinear activation function such as the logistic function which is used in this research. The reason for this requirement is that without nonlinearities, the network can be reduced to a single-layer network and lose the input-output relation properties of nonlinear networks; (2) the network must contain one or more hidden layers that can enable the network to handle complex tasks; (3) The network must exhibit a high degree of connectivity, determined by the weights of the network.

In a fully connected multilayer feedforward network, each neuron in one layer is connected by the multiplied weight of every neuron in the layer downstream to it. A bias is also associated with each of these weighted sums. Thus, in computing the value of each neuron in the hidden and output layers, the sum of the weighted sums and the bias must be calculated and fed into the activation function (as with the sigmoid function) in order to calculate the neuron's output.

The multilayer feedforward network can be represented as,

$$y(x) = \sum_{j=1}^{n_h} w_{2,j} v(w_{1,j}x + b_j), \quad (3.29)$$

where v is the activation function.

There are several common neuron activation functions that can be chosen, some of which are shown in Fig. 3.6. They are all critical to neuron information processing. The most widely used function is the sigmoid function, which is an “S” shaped function. The sigmoid function often refers to a special case of the logistic function which has a lower bound of zero and upper bound of 1, and a cross point with the y axis (0, 0.5). The mathematical formulation of the logistic function is

$$\text{logsig}(t) = \frac{1}{1+e^{-t}}, \quad (3.30)$$

where the e is the base of the natural logarithm.

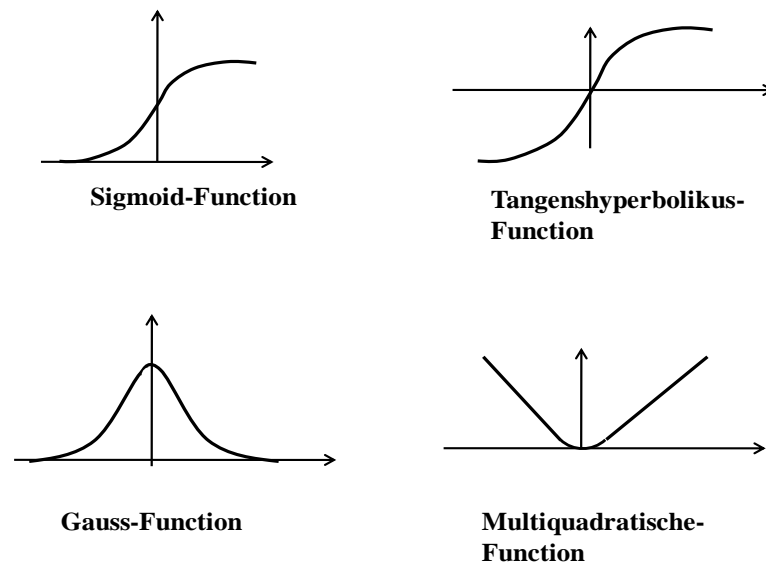


Fig. 3.6. Neuron activation functions.

Sigmoid functions are often used in neural networks to introduce nonlinearity in the model, and to make sure that certain signals remain within a specified range. A neural net element computes a linear combination of its input signals and applies a bounded sigmoid function to the result; this model can be seen as a "smoothed" variant of the classical threshold neuron. In simple terms as the gain increases, the slope of the activation function of the neurons decreases.

One reason for its popularity in neural networks is the sigmoid function satisfies the following property:

$$\frac{d}{dt} \text{logsig}(t) = \text{logsig}(t)(1 - \text{logsig}(t)). \quad (3.31)$$

This simple polynomial relationship between the derivative and itself is computationally easy to perform.

Inputs entering a neuron not only get multiplied by weights, they also get multiplied by the neurons transfer function. The sigmoid function is a typical neuronal

nonlinear transfer function that helps to make outputs reachable. The nonlinearity is significant. If the transfer function was linear, each of the neuronal inputs would get multiplied by the same proportion during training. This would cause the entire system to "drift" during training runs. The system would lose outputs it has already tracked while attempting to track new outputs. The non-linearity in the system helps to isolate specific input pathways.

Moreover, the use of the logistic function is biologically motivated, since it attempts to account for the refractory phase of real neurons.

After choosing the logistic function as the activation function, the multilayer network can be stated as:

$$y(x) = \sum_{j=1}^{n_h} w_{2,j} s(w_{1,j}x + b_j) = \sum_{j=1}^{n_h} \frac{w_{2,j}}{1 + e^{-(w_{1,j}x + b_j)}}. \quad (3.32)$$

The neural network learns and evaluates the weights by the way of backpropagation.

Backpropagation, which is usually applied to neural networks, is a very powerful tool, especially in the areas of pattern recognition, dynamic modeling and faults diagnosis. This algorithm is based on error-correction learning. The core principle of backpropagation is to calculate the derivatives exactly and efficiently within a system. The theorem underlying backpropagation is the chain rule for ordered derivatives.

Similar to the algorithm of LMS, backpropagation takes the partial derivatives of the error or mean square error and applies them to each of the weights, starting from the output layer to the hidden layer weights, and then the hidden layers to the input layer weights. As it turns out, this is necessary since changing these sets of weights requires

that we know the partial derivatives calculated in the layer downstream. This is the reason this algorithm is called the "back propagation algorithm."

3.6 Chapter Summary

In this chapter, a method for calculating Lyapunov exponents from time series data is described. The method mainly includes three parts: phase space reconstruction, a multilayer neural network, and an evaluation of the Lyapunov exponents using the QR decomposition. In the next chapter, this method is applied to several example problems in order to compute the Lyapunov exponent.

CHAPTER IV

SIMULATION RESULTS

4.1 Introduction

In this chapter, the previously developed method for calculating the LLE is applied to several simple chaotic systems: Lorenz and Rössler for continuous flow and Hénon for discrete map. For the Lorenz system, the effects of changing parameters and adding noises to the inputs are analyzed. In order to study chaotic dynamics, the Rayleigh number, a parameter of the Lorenz system is changed in a large scale. The estimated results are compared with the theoretical values which are directly calculated from the equations of the systems. All of the simulation results are included in this chapter.

4.2 The Lorenz Attractor

4.2.1 Introduction of the Lorenz Attractor and Its Phase Space Reconstruction

The Lorenz attractor is very important in the field of non-linear mathematics. It was firstly introduced by Edward Lorenz in 1963. The equations are based on a simplified model of atmospheric convection rolls. The three coupled differential equations are

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= Rx - y - xz, \\ \frac{dz}{dt} &= xy - \beta z,\end{aligned}\tag{4.1}$$

where x , y , z are the three states of the system and σ (Prandtl number), R (Rayleigh number), β are three parameters characterizing the particular properties of the flow. Different parameters can change the appearance of the Lorenz attractor. Analysis of the effects of changing these parameters is discussed in Section 4.2.3. In this example, the three parameters are set to $\sigma = 10$, $\beta = 8/3$ and $R = 28$, and the Lorenz system exhibits chaotic behavior. The time series plots for each variable and the three-dimension appearance in phase space of this Lorenz attractor are shown in Fig. 4.1 and Fig. 4.2, respectively.

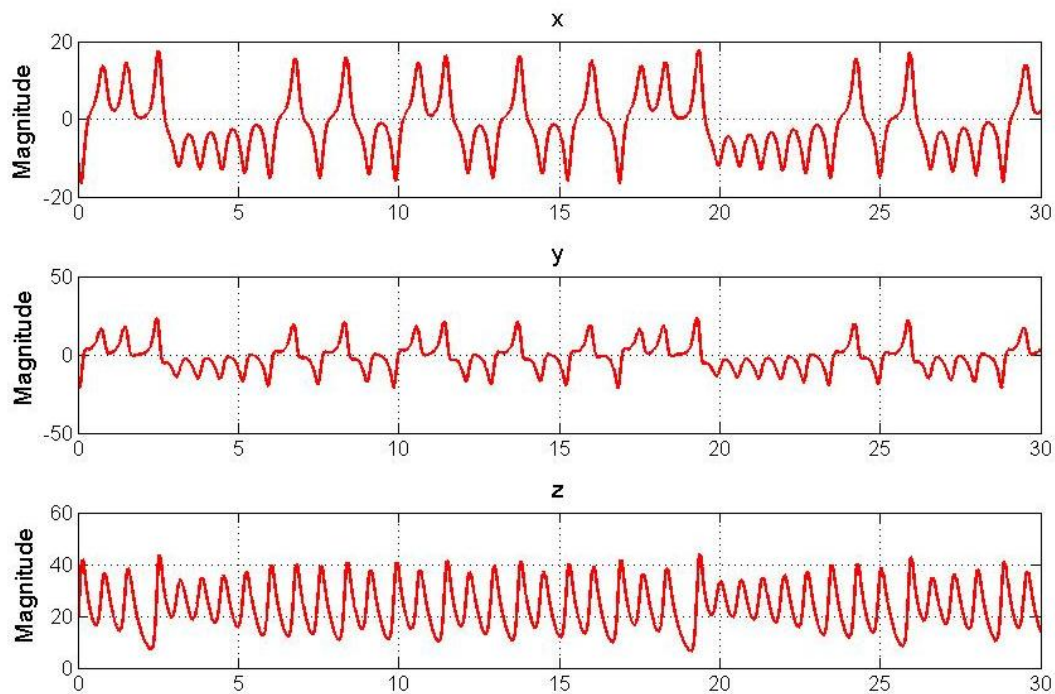


Fig. 4.1. Time series plots for the three states of the Lorenz system.

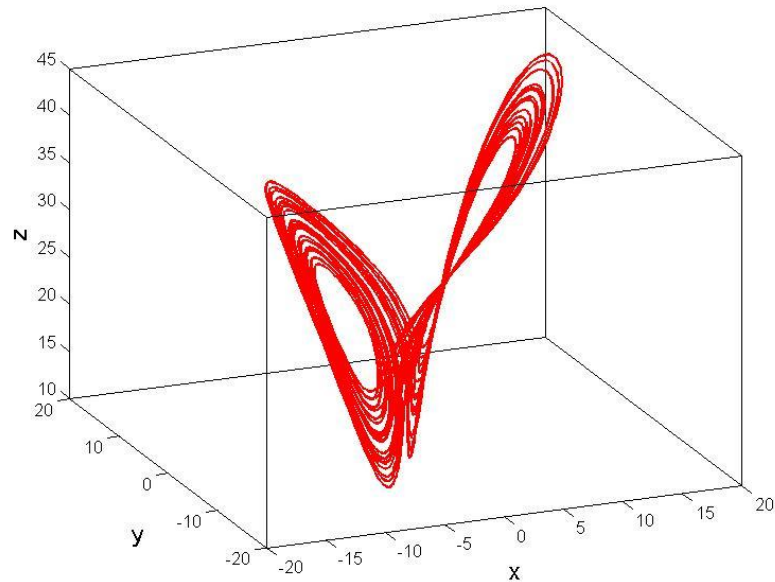


Fig. 4.2. Three-dimensional plot of the Lorenz system.

The trajectory is generated by solving the ordinary differential equations using the Runge-Kutta method with 0.001 second max step. The initial conditions for the three variables are (1, 0, 0). The data is extracted with 0.01 second sampling time. Trajectory recording begins after 20 seconds to allow the states to be on the attractor. Then, 1000 seconds of data are recorded and saved in a file.

Assuming only the states are available and the other information is unknown, the phase space can be reconstructed by the methods stated in the previous chapter. The time delay and the embedding dimension are computed by applying the Average Mutual Information and the False-Nearest-Neighbors method, respectively. Fig. 4.3 shows the mutual information based on the data file. The x-axis is for the time delay with units of second and the y-axis is for the Average Mutual Information of the original data and the time-delayed data.

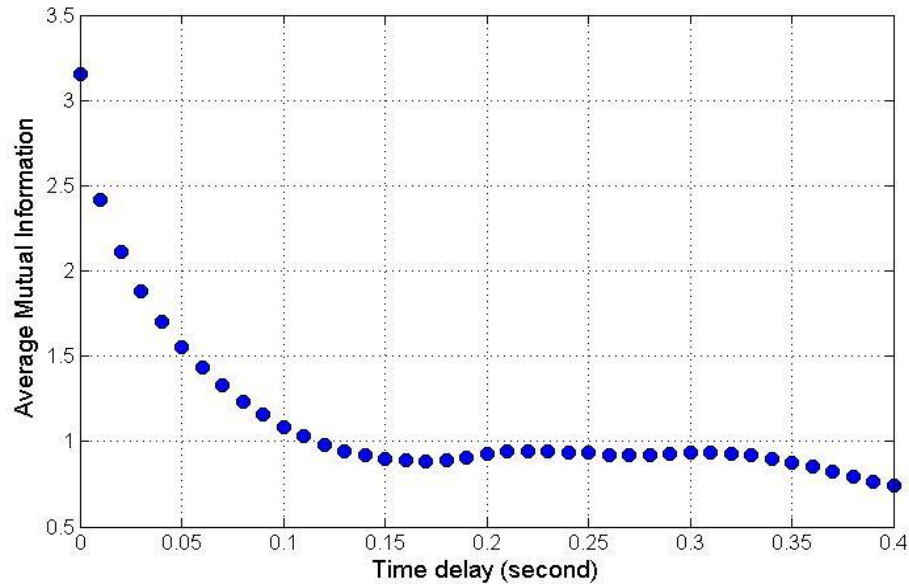


Fig. 4.3. Average mutual information for the Lorenz system.

From the plot of the Average Mutual Information, the first local minimum value is obtained as the time delay around 0.16 second. Actually, the time delay does not need to be the exact theoretical value. The effect of the time delay on calculating the LLE of this Lorenz system is further discussed in later section. In fact, use of 0.1 second time delay provides a slight improvement.

After obtaining the sufficiently accurate time delay, the False Nearest Neighbors method with 0.01 of the fraction R is applied to compute the embedding dimension. The result is three dimensions for the Lorenz system. The plot of the percentage frequency of the false nearest neighbors through the embedding dimension is shown in Fig. 4.4. If the time delay of 0.16 second is used, the plot is similar, and the embedding dimension is still three.

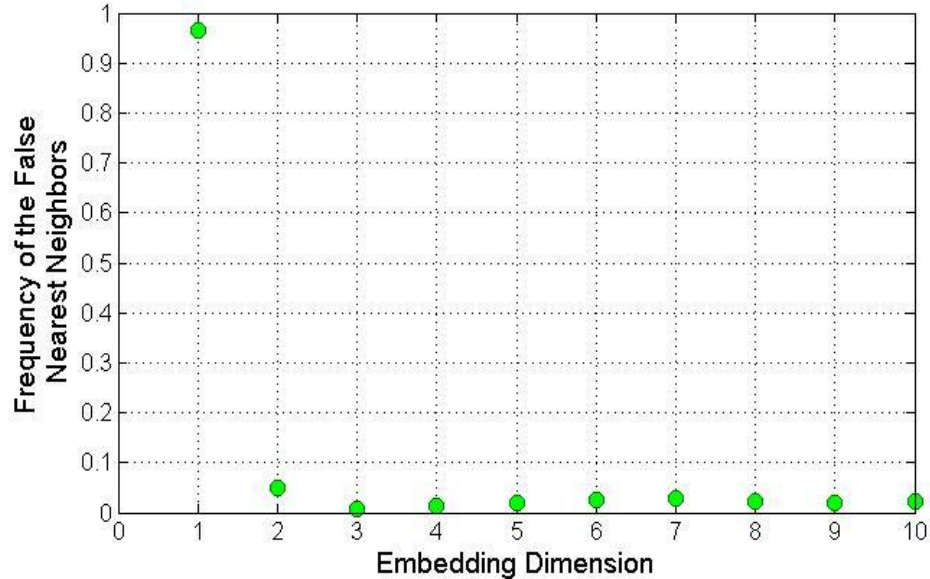


Fig. 4.4. The result of applying the False Nearest Neighbors method to the Lorenz system.

After the determining the time delay and the embedding dimension, the phase space can be reconstructed from the time series data, and the LLE can be calculated through the reconstructed phase space.

4.2.2 The Largest Lyapunov Exponent Calculation for the Lorenz System

The Lorenz system with parameters $\sigma = 10$, $R = 28$, $\beta = 8/3$, has the theoretical LLE of 0.9022. In order to test the calculation method based on the multilayer feedforward neural network, the hidden neurons and the number of lags must be specified. The number of hidden neurons in each particular case should be determined by taking into account the complexity of the problem. The more complicated the mapping, the more hidden neurons that are required. Unfortunately, there is no universal rule that can be applied. Usually, the number of the hidden neurons is determined by

experience and experiment. In general, if the number of the hidden neurons is too large, the calculation will be very slow. Conversely, if the number of the hidden neurons is too small, the neural network is unable to accurately produce the desired signals. In addition, too many neurons can contribute to over fitting, in which all training points are well fit, but the fitting curve takes wild oscillations between these points. The number of 100 hidden neurons is arbitrarily selected in this test. The number of lags uses the embedding dimension, which is estimated by using the False Nearest Neighbors method. The previously obtained result of the embedding dimension is three. Consequently, the Lorenz system is tested with a neural network using three lags.

In order to obtain a good evaluation result, a reasonable tested data size or the time length must be chosen, and the optimal time delay needs to be verified. Larger data size can obviously produce a better evaluation result. However, much more time and memory resources are needed. If the sampling time is chosen as 0.16 second as the result of calculating average mutual information method, a group of data sets with different time lengths (from 50 seconds to 400 seconds) are applied.

In addition, the initial values of the weights and the biases of the neural network are randomly chosen since it is assumed that nothing is known about the model before training. Consequently, errors can be induced in the neural network model. Several statistical methods such as multi-testing and averaging are used to reduce the errors.

In the test, one data set is used both for training the neural network and estimating the largest Lyapunov exponent. The evaluation is done ten times repeatedly with different randomly initial weights and biases to obtain ten values of the LLE. Then,

the average value is attained, and the plot of the average value for 50-400 seconds is shown in Fig. 4.5.

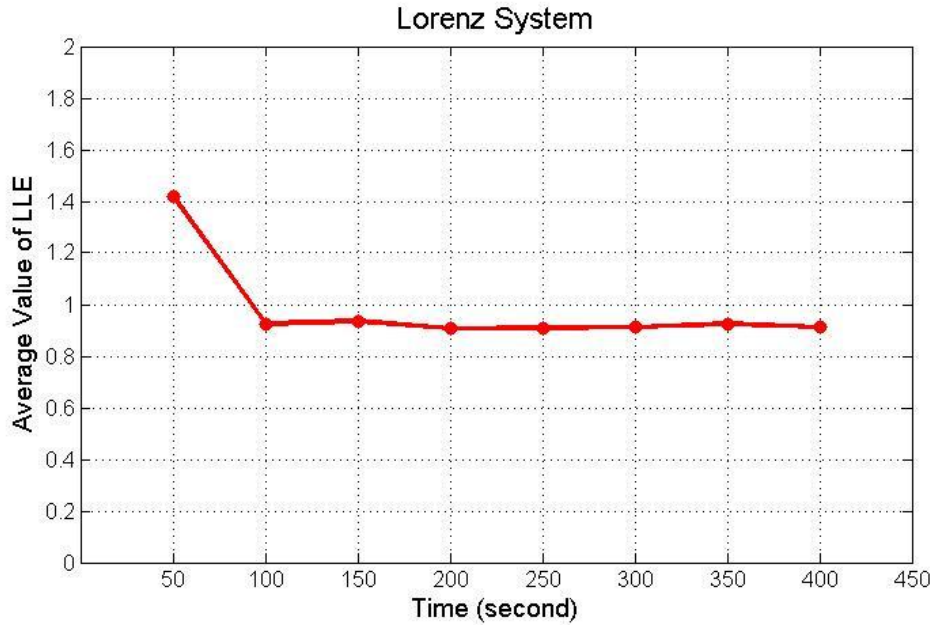


Fig. 4.5. Calculated value of the LLE for the Lorenz system.

From the plot, the calculation drops sharply between 50 to 100 seconds. Then the evaluated value stays close to the theoretical value. In order to obtain a better sense of the evaluation results, the relative error and the standard deviation of the LLE are calculated as follows:

$$Relative\ error = \left| \frac{\overline{LLE} - \widehat{LLE}}{\widehat{LLE}} \right| \quad (4.2)$$

where \overline{LLE} is the average of the multi-testing LLE, \widehat{LLE} is the theoretical value of LLE.

The standard deviation σ is computed as:

$$\sigma = \sqrt{\frac{\sum(LLE_i - \overline{LLE})^2}{n-1}} \quad (4.3)$$

where LLE_i is the i th calculation for the LLE.

Fig 4.6 and Fig. 4.7 show the results, and the trends are similar to the average value. To satisfy the small relative error and low standard deviation, a choice of 200 seconds is reasonable. The corresponding result values are listed in Table A.1 in Appendix A.

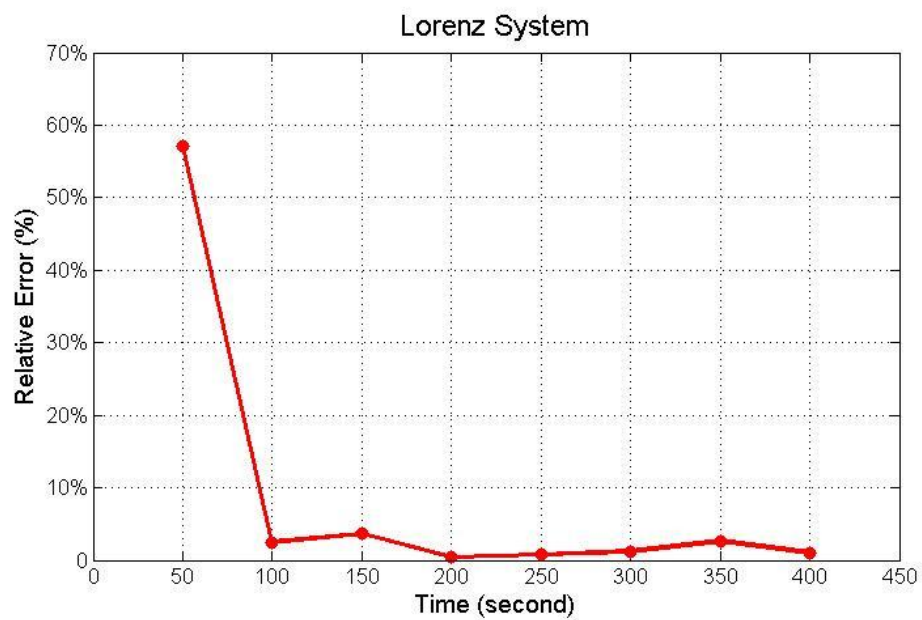


Fig. 4.6. Relative error of the LLE for the Lorenz system.

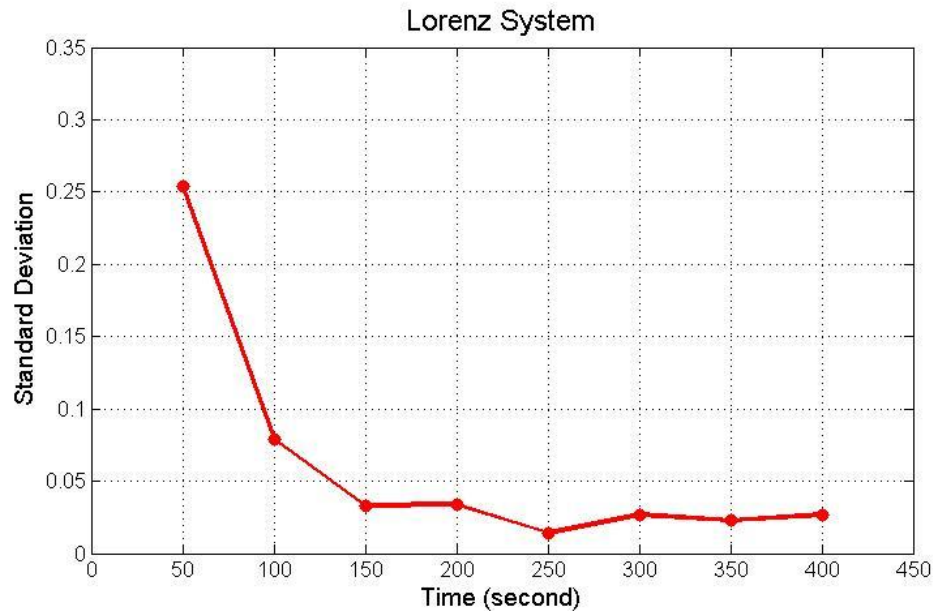


Fig. 4.7. Standard deviation of the LLE for the Lorenz system.

After determining the time length of the time series data, the time delay is further verified. The method of Average Mutual Information can give us a general time delay. However, it may not be the best. In order to verify the optimal time delay, the data sets with different time delays are tested. The time delay varies from 0.01 to 0.2 second. When the time delay is changed, the embedding dimension always needs to be evaluated again by the False Nearest Neighbors method. But for this test, the reevaluated embedding dimension is still three.

Then, the LLE can be evaluated ten times for each case. The average value, the relative percent error and the standard deviation are calculated in the same way as the previous test. The results are shown in Figs. 4.8-4.10, respectively.

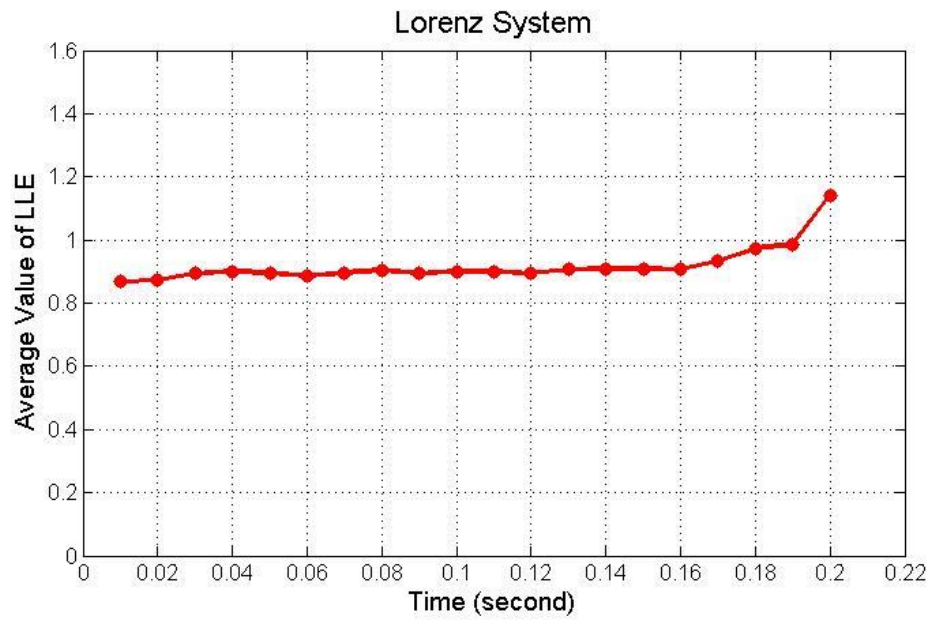


Fig. 4.8. Average LLE for the Lorenz system with different time delay.

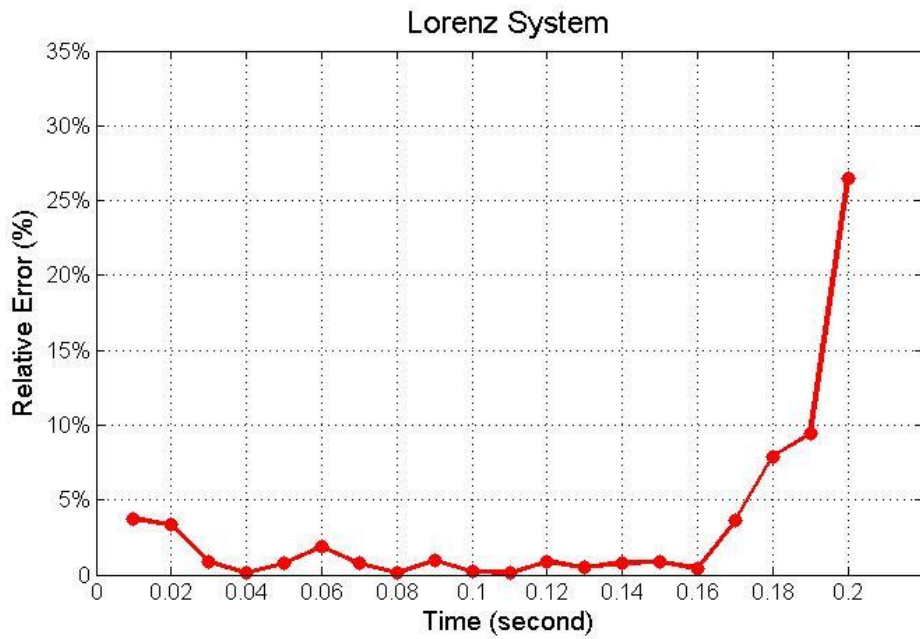


Fig. 4.9. Relative error of the LLE for the Lorenz system with different time delay.

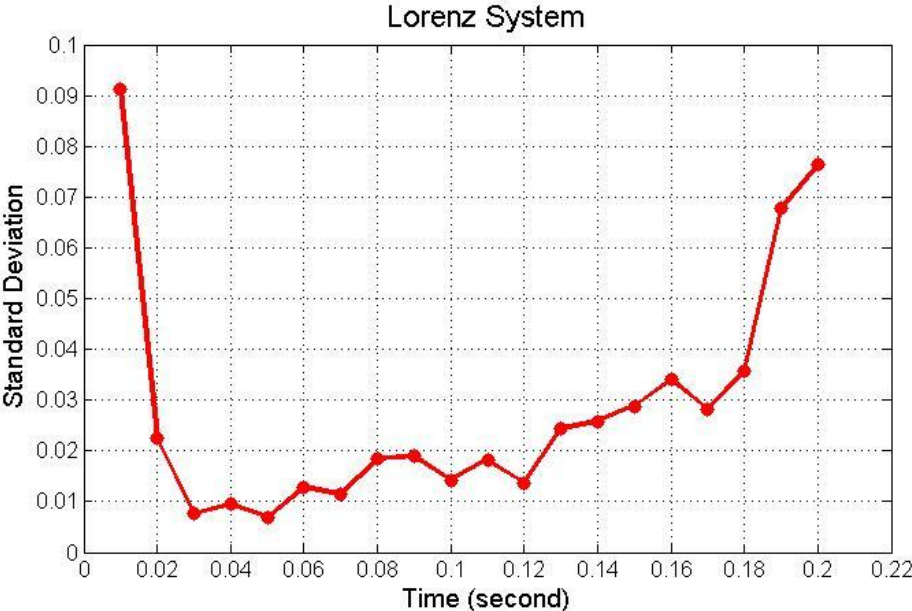


Fig. 4.10. Standard deviation of the LLE for the Lorenz system with different time delay.

From Fig. 4.8 to Fig. 4.10, the best evaluation is obtained at a range of the time delay, but not only exact one. The time delay of 0.1 second is used in the later tests, because the evaluation results by using this time delay is better than the results of using 0.16 second. The comparison of these two time delays is displayed in Table 4.1.

Table 4.1. Comparison between the largest Lyapunov exponent evaluation results with 0.1 and 0.16 seconds time delays.

Time Delay (Sec)	0.10	0.16
LLE	0.9002	0.9060
LLE Relative Error (%)	0.22%	0.42%
LLE Standard Deviation	0.0143	0.0340

When all of the sets are fixed, a new question arises. How often should the neural network be updated? To answer this question, a new experiment is performed. For the test, two time series data sets are extracted from the Lorenz system. The second set is

from 0 to 9000 seconds following the first set. The first set is for training the neural network, and the second set is for estimating the LLE. After ten times of calculations for each case, the average value, the relative percent error, and the standard deviation can be obtained. The results are summarized in Table 4.2.

Table 4.2. Summary of the results for different time gaps between the data set for training the neural network and the data set for evaluating the LLE.

Time Gap (Sec)	0	1000	2000	3000	4000	5000	6000	7000	8000	9000
LLE	0.9078	0.9162	0.9268	0.9102	0.9103	0.9149	0.9041	0.9324	0.9061	0.9050
LLE Relative Error (%)	0.0062	0.0155	0.0272	0.0088	0.0090	0.0140	0.0021	0.0334	0.0043	0.0031
LLE Standard Deviation	0.0081	0.0197	0.0158	0.0197	0.0094	0.0156	0.0213	0.0172	0.0222	0.0137

From the table, the calculated value is always near the theoretical LLE, and the relative percent error and the standard deviation remain small. The results prove that the trained neural network can be used to evaluate the LLE for a long time without updating the neural network, at least 9000 seconds (2.5 hours) for the Lorenz system case.

Up to now, most features for the evaluation of the LLE based on the time series data from the original Lorenz system are verified, and reasonable results are obtained. But in reality, the parameters of a dynamical system possibly change for several reasons, such temperature, humidity, background noise, and so on. Thus, the test of applying the method on the system with changed parameters is valuable. First, the evaluation is performed without retraining the neural network. In this test, the neural network trained by the original Lorenz system data is still used. The parameters of the Lorenz system are changed to get a new time series data, which is used to perform the LLE evaluation. The parameters σ , R , β are each changed $\pm 5\%$ and $\pm 10\%$. The theoretical value and the

calculated value of LLE are shown in Fig. 4.11. The selected parameters for each case and the corresponding calculation results are listed in Table A.2 in Appendix A.

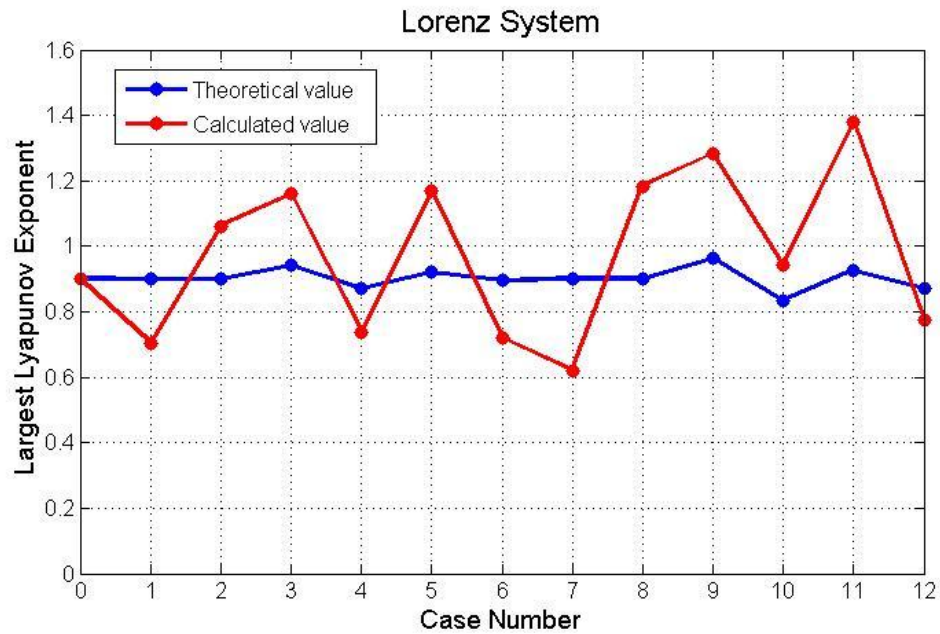


Fig. 4.11. The theoretical value and the calculated value of the LLE for the 5% or 10% parameters changed Lorenz system without retraining the neural network.

Furthermore, the neural network is trained by the data from the system with 3% parameter changes, and the data from the system with 5% parameter changes is used to evaluate the LLE. Fig. 4.12 shows the results. Table A.3 in Appendix A gives the selected parameters for each case and the calculation results.

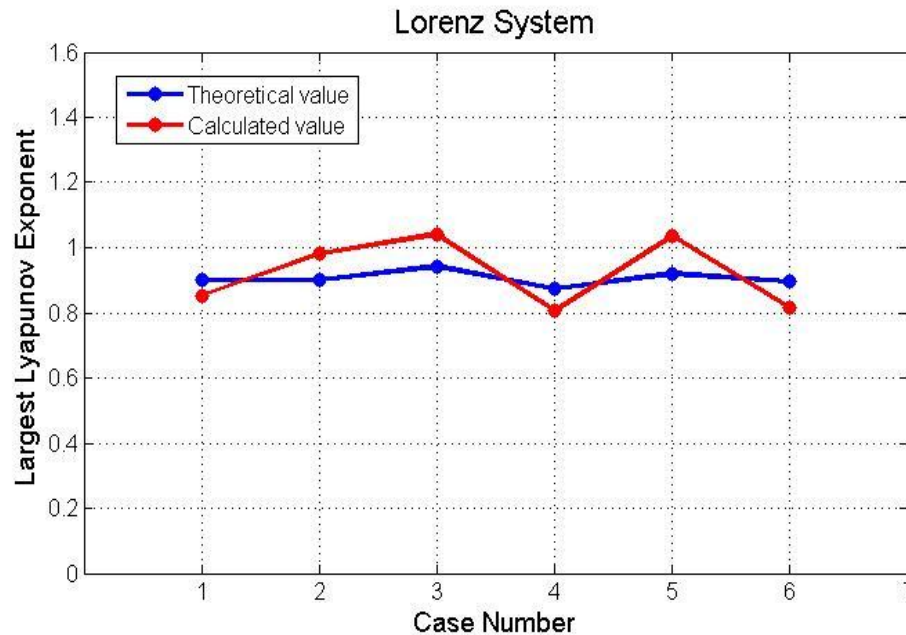


Fig. 4.12. The theoretical value and the calculated value of the LLE with the neural network trained by 3% parameters changed Lorenz system data.

From Fig. 4.11 and Fig. 4.12, the application of the networks trained by the original Lorenz system onto the data from the system with 5% or 10% parameter changes can find the change directions of the LLE. However, the changes are obviously enlarged. The networks trained by the data with 3% parameter changes can lead the same trends but are closer to the theoretical values. In addition, different parameters have different effects on the LLE evaluation. The parameter σ has little effect on the theoretical value of the LLE. But the parameters R and β affect the LLE relatively more. Besides, the change of the LLE becomes larger when the parameter changes are larger.

Next, the evaluation with retraining the network is performed. The time series data obtained from one Lorenz system is used both to train the neural network and to evaluate the largest Lyapunov exponent. Fig. 4.13 and Fig. 4.14 show the results for 5%

and 10% parameters changed Lorenz systems. The corresponding values are shown in Table A.4 and Table A.5 in Appendix A.

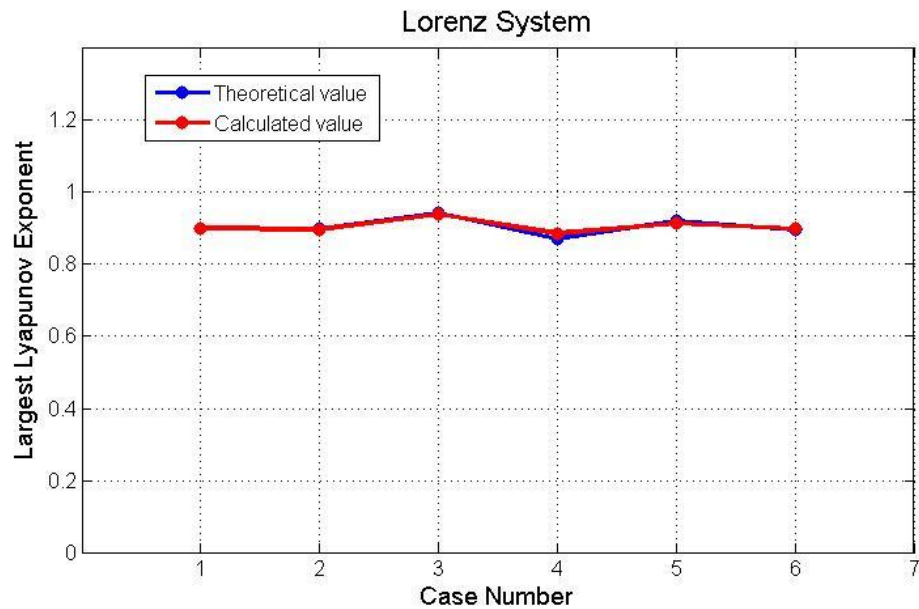


Fig. 4.13. The theoretical value and the calculated value of the LLE for 5% parameters changed Lorenz system with retraining neural network.

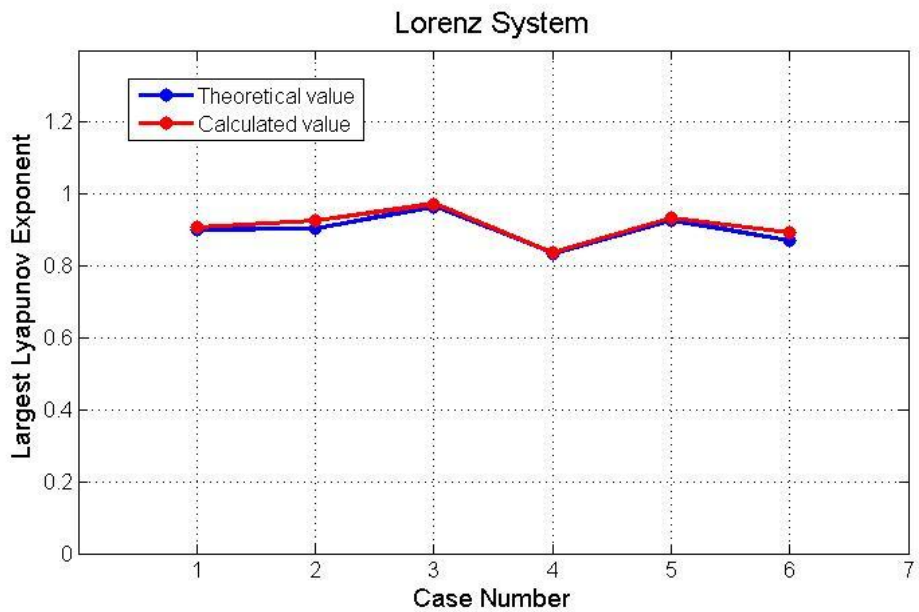
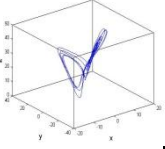
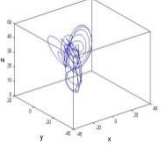
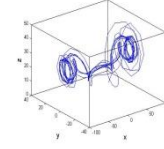
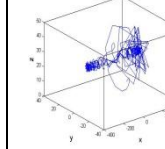
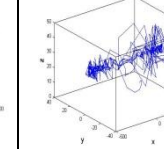
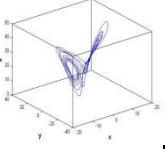
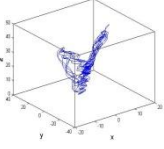
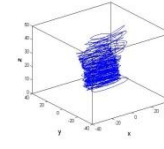
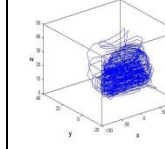
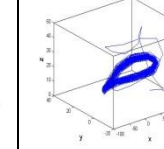


Fig. 4.14. The theoretical value and the calculated value of the LLE for 10% parameters changed Lorenz system with retraining neural network.

From Fig. 4.13 and 4.14, using the data from the same system both for training neural networks and evaluating LLE can always attain the results close to the theoretical values. So in reality, the neural network must be updated regularly, because the parameters of the dynamical system have possibly changed in a short time.

In the real world, systems can be also disturbed by some external excitation, of which Sinusoidal function is a common formulation. In order to check such effects on LLE, a Sinusoidal function as input is added to the first function of the original Lorenz system. The amplitude of the input is varied from 0.1 to 50 and the frequency is changed from 1 Hz to 10 Hz, respectively. The three-dimensional plots are shown in Table 4.3. According to this group of plots, if the amplitude of the Sinusoidal function is very small, the attractor is not significantly affected. When the amplitude is large, the shape of the attractor is seriously varied while the frequency is changing.

Table 4.3 Three-dimensional plots for Lorenz system with different Sinusoidal inputs.

Amplitude \ Frequency	0.1	1	10	30	50
1 Hz					
10 Hz					

From Table 4.3, the shape is changed from the original Lorenz attractor to some other attractors when the amplitude increases. Then the input of Sinusoidal function with

the amplitude 1, 5, 10 and the frequency 1 to 10 Hz are induced to the Lorenz system for LLE evaluation. Tables A.6-A.8 in Appendix A list the theoretical values, numerical average values, the relative percent errors, and the standard deviation of the LLE. Figs. 4.15-4.17 compare the theoretical values and the numerical results.

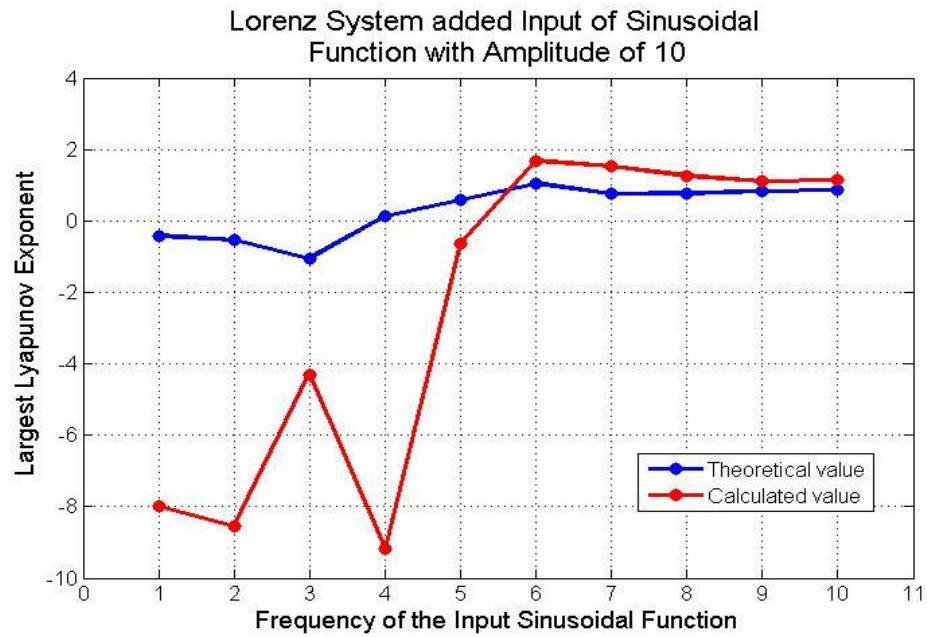


Fig. 4.15. The theoretical value and the numerical value of the LLEs for the Lorenz system added Sinusoidal function input with the amplitude of 10.

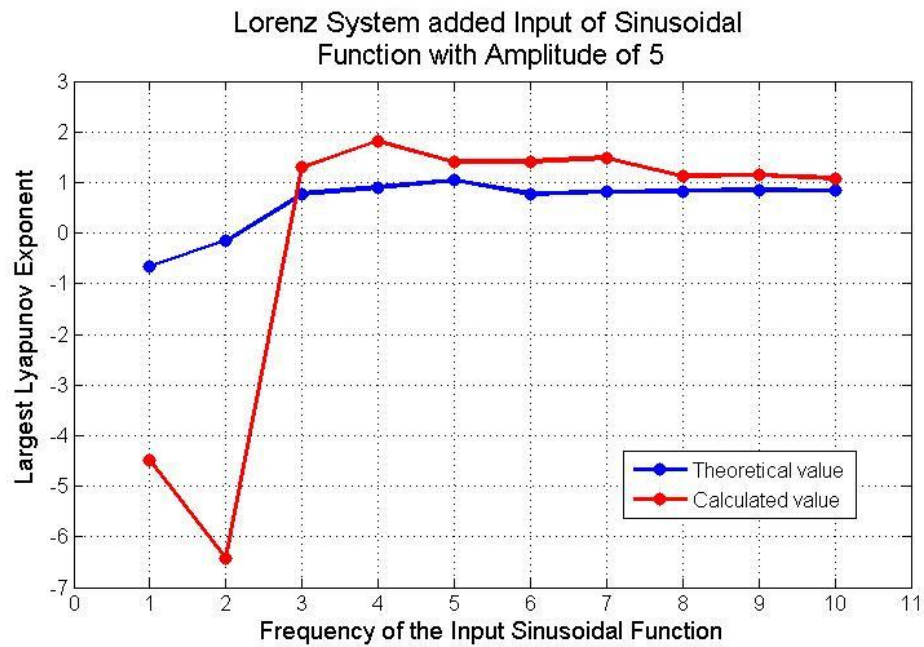


Fig. 4.16 The theoretical value and the numerical value of the LLEs for the Lorenz system added Sinusoidal function input with the amplitude of 5.

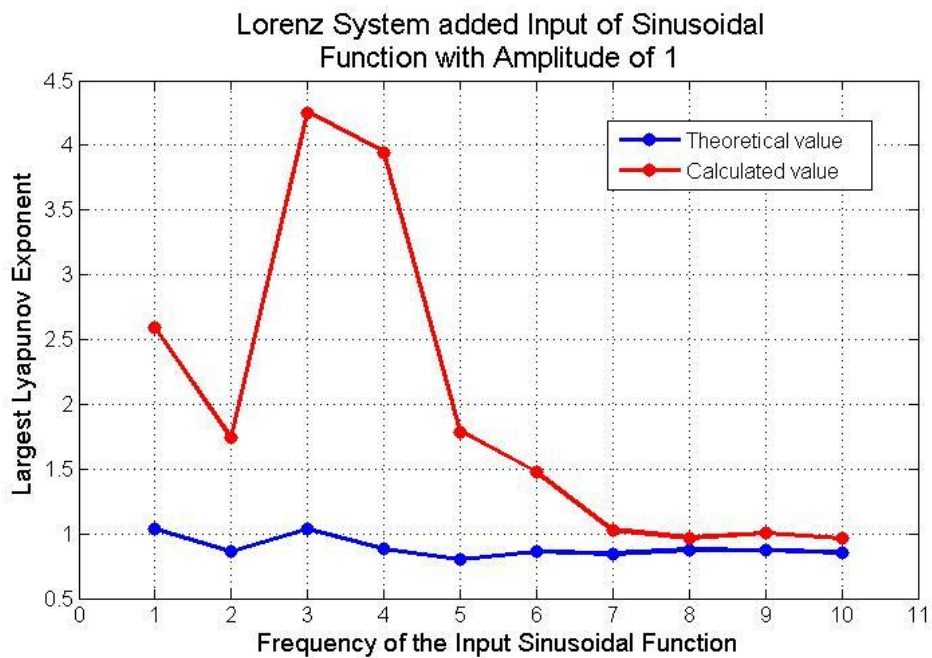


Fig. 4.17. The theoretical value and the numerical value of the LLEs for the Lorenz system added Sinusoidal function input with the amplitude of 1.

In the real world, the systems are always accompanied with noise. The effect of noise sometimes cannot be ignored. In this research, the Gaussian white noise is added to the original Lorenz system. A scalar SNR which specifies the signal-to-noise ratio per sample is used to evaluate the noise large or small. It is expressed as

$$\text{SNR}(dB) = 10\log_{10}\left(\frac{S}{N}\right) \quad (4.4)$$

where S is the input signal power, and N denotes the noise power.

SNR is valued 4.7712, 6.9897, 10, 13.0103, 20, corresponding S/N equals to 3, 5, 10, 20, 100 respectively. 5 new groups of data from these systems with added noise are obtained.

First, the neural network trained by the original Lorenz system data is used, and the data from the system with added noise is used to evaluate the LLE. The results are recorded in Table A.9. in Appendix A.

Next, the noise-added data is used to both train the neural network and evaluate the LLE. The results are recorded in Table A.10. in Appendix A.

Fig. 4.18 compares the theoretical LLE for the original Lorenz system and the estimated LLE for the added-noise system with and without retraining neural network.

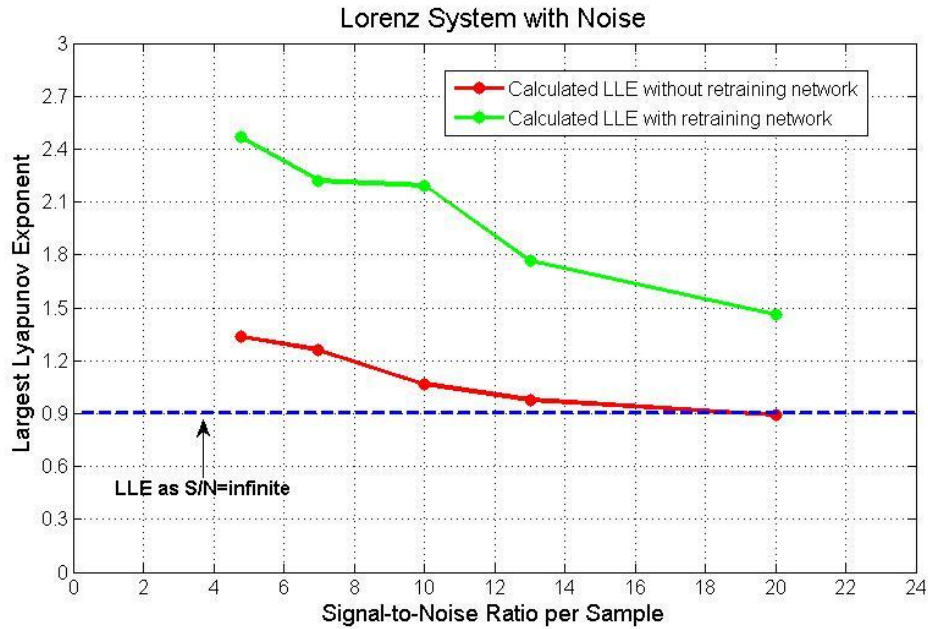


Fig. 4.18. The theoretical LLE for the original Lorenz system and the numerical LLEs for the noise-added Lorenz system.

From the plot, when the noise is small, the evaluation result without retraining network can match the theoretical LLE. However, the evaluation with retraining network is different. Because the theoretical value for the system with noise is changed and unknown.

4.2.3 Chaotic Dynamics of the Lorenz System

The Lorenz system tested in Section 4.1 is still used in this section and the system is rewritten below as

$$\begin{aligned}
 \frac{dx}{dt} &= \sigma(y - x) , \\
 \frac{dy}{dt} &= Rx - y - xz , \\
 \frac{dz}{dt} &= xy - \beta z .
 \end{aligned}
 \tag{4.5}$$

The parameters σ , R and β of Lorenz system can change its behavior.

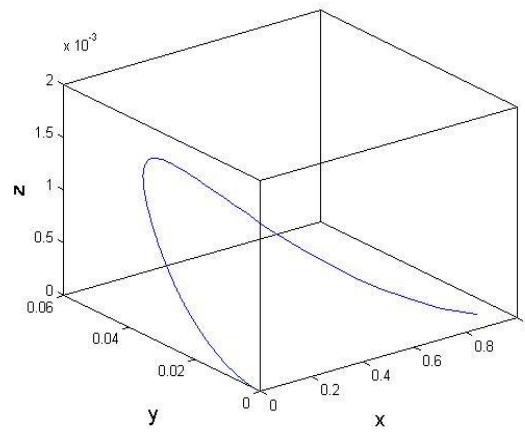
Usually $\sigma = 10$ and $\beta = 8/3$. For the case of fixed σ and β , the shape of this system varies with the different values of R . The equilibrium conditions are separated as:

- (1) When $R < 1$, the system is stable at point $(0, 0, 0)$.
- (2) When $1 < R < R_c$, the system is still stable, but there are three stable equilibrium points which are $(0, 0, 0)$, $(\sqrt{\beta(R-1)}, \sqrt{\beta(R-1)}, R-1)$ and $(-\sqrt{\beta(R-1)}, -\sqrt{\beta(R-1)}, R-1)$ where $R_c = \sigma \frac{\sigma+\beta+3}{\sigma-\beta-3}$.

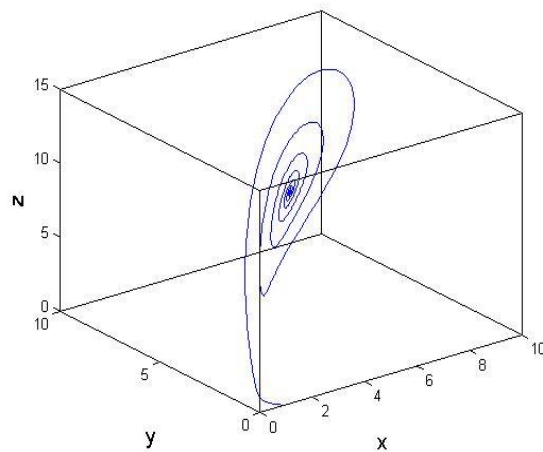
- (3) $R > R_c$, the system goes into chaos. Small change of parameter R or small change of initial condition may change the appearance of the attractor severely.

Fig. 4.19 shows 3-D plots of some examples for the three different conditions. Fig. 4.19 (a) shows the behavior of the Lorenz system with $R= 0.5$, which belongs to condition (1) above; Fig. 4.19 (b) is for the system with $R= 10$, which belongs to the condition (2) above; and the behavior of the system with $R= 28$, which belongs to condition (3) above, is shown in Fig. 4.19 (c).

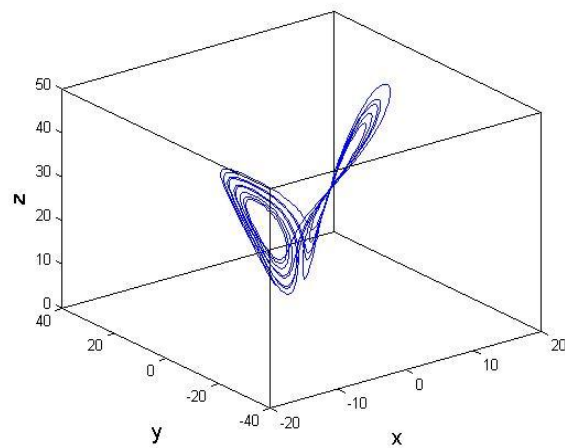
The Lorenz system with different values of the parameter R is tested by the method of calculating the LLE. The parameters σ and β are fixed, and the parameter R is varied from 10 to 30. Fig. 4.20 (a) shows the plot of the LLE versus R for the system without noise. Then, three different Gaussian white noise levels (SNR=20, 13, 10) are added to this system, respectively. The results are shown in Fig. 4.20 (b)-(d).



(a)

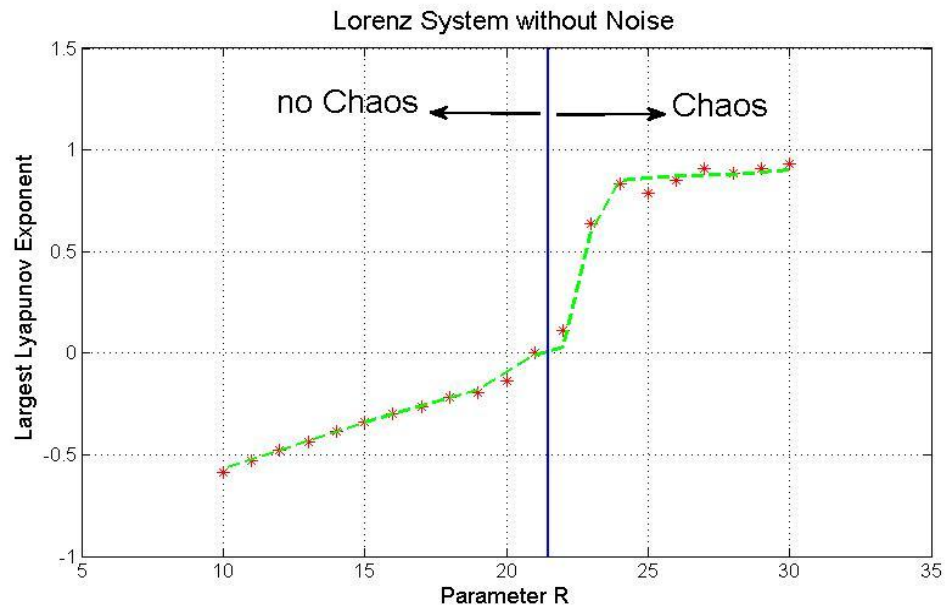


(b)

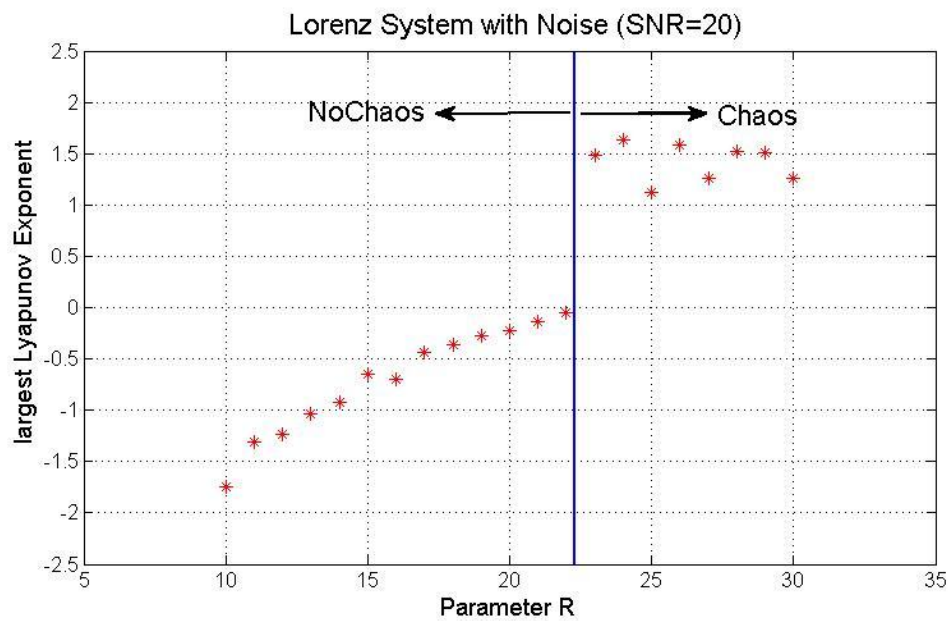


(c)

Fig. 4.19. The 3-dimensional plot of the Lorenz system with the parameters (a) $\sigma=10$, $R=0.5$, $\beta=8/3$; (b) $\sigma=10$, $R=10$, $\beta=8/3$; (c) $\sigma=10$, $R=28$, $\beta=8/3$.

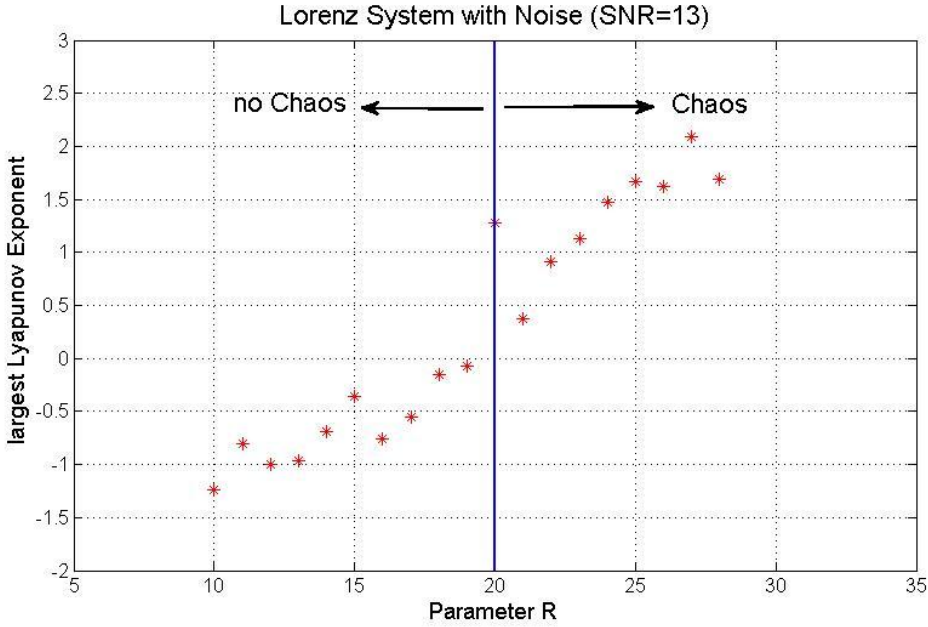


(a)

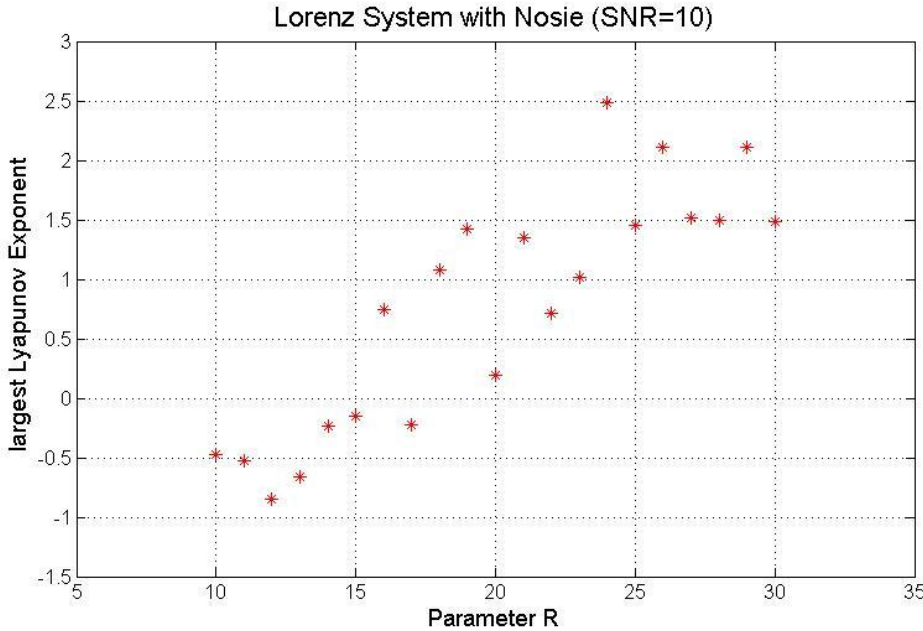


(b)

Fig. 4.20. Largest Lyapunov exponent evaluation for (a) the non-noise Lorenz system; (b) the Lorenz system added the noise with SNR=20; (c) with SNR=13; (d) SNR=10.



(c)



(d)

Fig. 4.20. Continued.

Fig. 4.20 (a) gives a clear relationship of the largest Lyapunov exponent and the parameter R . The evaluated value matches the theoretical value. The results can distinguish chaos and no chaos. Fig. 4.20 (b)-(d) show the effect of noise on the evaluation results. When the noise is low, the evaluated LLE can still indicate the occurrence of chaos like the no-noise system. But as the noise increases, the relationship pattern for the parameter R and the LLE is not very clear. Thus when the noise is not very large, the method for calculating the LLE has a certain level of robustness, and is capable of detecting chaos in a dynamical system under noise background.

4.3 The Hénon Map

The Hénon map is a discrete map, which is one of the most studied examples of dynamical system models for chaos. It is a two-dimension iterated map. Its chaotic solutions are proposed as a simplified model of the Poincare map for the Lorenz model by the French astronomer Michel Hénon in 1976. The Poincare map is known as a two dimensional map which is extracted from an attractor of a continuous dynamical system with more than two dimensions. The Hénon map is expressed by the coupled equations

$$x(n + 1) = 1 - ax(n)^2 + y(n), \quad (4.6)$$

$$y(n + 1) = bx(n),$$

where a and b determine the map. For different values, the map may be chaotic, intermittent, or periodic. When $a=1.4$ and $b=0.3$, the map is Canonical Hénon map which exhibits chaotic behavior. The two dimensional attractor is shown in Fig. 4.21.

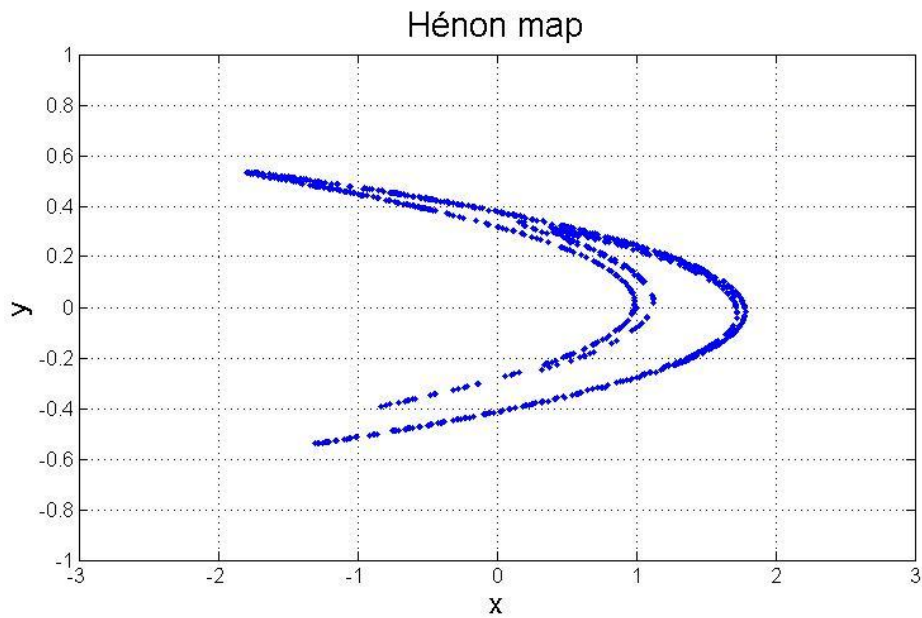


Fig. 4.21. The appearance of the attractor of Hénon map.

There are two Lyapunov exponents for the Hénon map due to the two dimensions. The theoretical values of the two exponents for Canonical Hénon map are 0.419217 and -1.623190 respectively. The discrete-time series data of the Hénon map can be obtained by an iteration process. With the time delay set to 1, the embedding dimension can be evaluated by the method of False Nearest Neighbors (FNN). The result of the dimension is 2. The resulting plot of the FNN method, the fraction R versus the embedding dimension, is shown in Fig. 4.22.

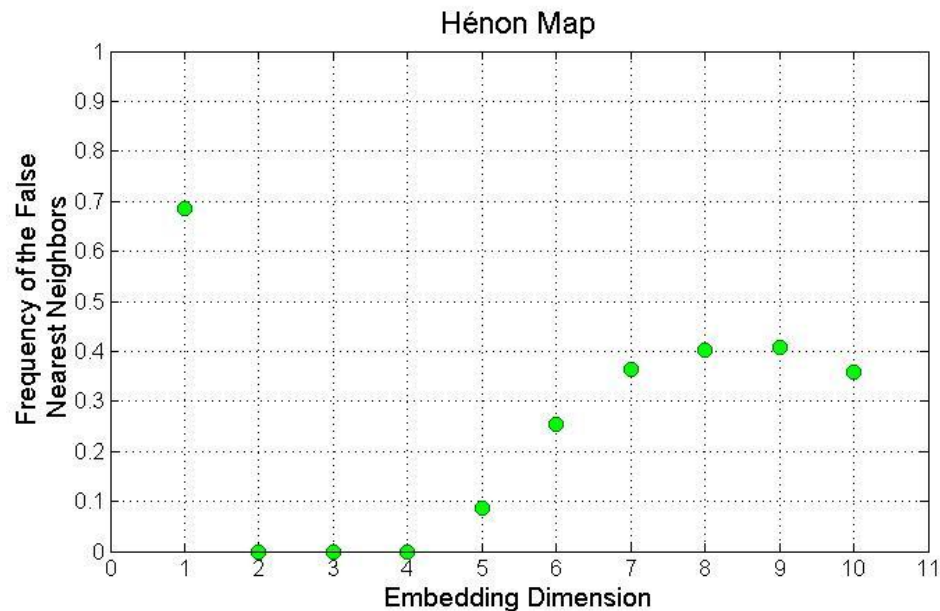


Fig. 4.22. The result of applying the False Nearest Neighbors method to the Hénon map.

Because the Hénon map is a two-dimensional map and the system is much simpler than Lorenz, the number of hidden neurons needn't be large. Three hidden neurons are enough for this multilayer neural network to perform well, and the numerical results are: $LE1 = 0.4195$ and $LE2 = -1.6245$, which are very close to the theoretical values. Actually, Wolf's method introduced in Chapter 3 can also provide similar results.

4.4 The Rössler Attractor

The Rössler attractor is not a famous attractor. However, it is a rather nice attractor which draws a nifty picture. Rössler systems, which are a series of prototype systems for ordinary differential equations in three-dimensional phase space, were originally introduced by Otto Rössler in the 1970s.

Rössler was inspired by the geometry of flows in three dimensions and, in particular, by the reinjection principle [35]. This principle is based on the feature of relaxation-type systems that often present a Z-shaped slow manifold in their phase space. On this manifold, the motion is slow until an edge is reached, whereupon the trajectory jumps to the other branch of the manifold. This feature allows not only for periodic relaxation oscillations in dimension two (see Fig. 4.23(a)), but also for higher types of relaxation behavior (see Fig. 4.23(b)) as noted by Rössler (1979a). In dimension three, the reinjection can induce chaotic behavior if the motion is spiraling out on one branch of the slow manifold.

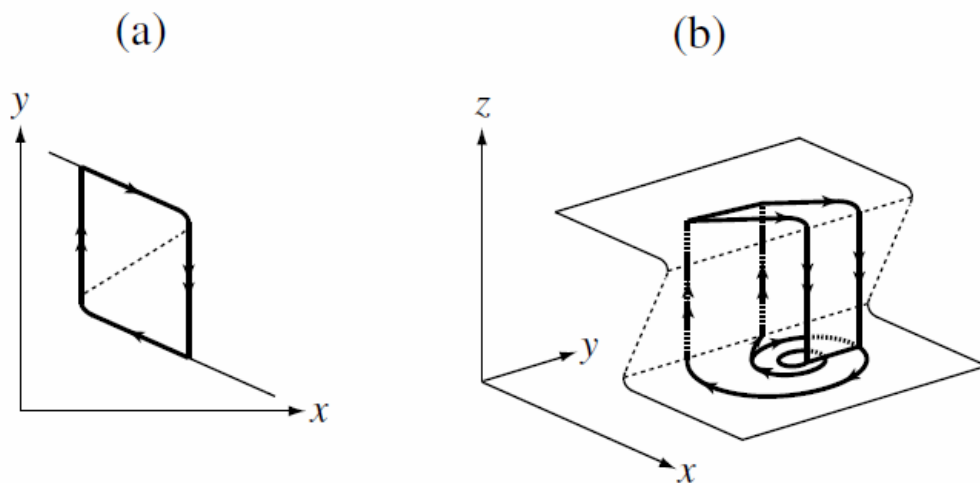


Fig. 4.23. Illustration of the reinjection principle between the two branches of a Z-shaped slow manifold allowing (a) periodic relaxation oscillations in dimension two and (b) higher types of relaxation behavior in dimension three [35].

In this way, Rössler invented a series of systems, the most famous of which is Rössler 1979a which is formed with a series of Navier-Stokes-like equations, namely:

$$\begin{aligned}\frac{dx}{dt} &= -y - z, \\ \frac{dy}{dt} &= x + ay, \\ \frac{dz}{dt} &= b + z(x - c),\end{aligned}\tag{4.7}$$

where $a = 0.15$, $b = 0.20$, $c = 10.0$ and the initial conditions are $x(0) = 1$, $y(0) = 1$, $z(0) = 1$. This attractor is shown in Fig. 4.24.

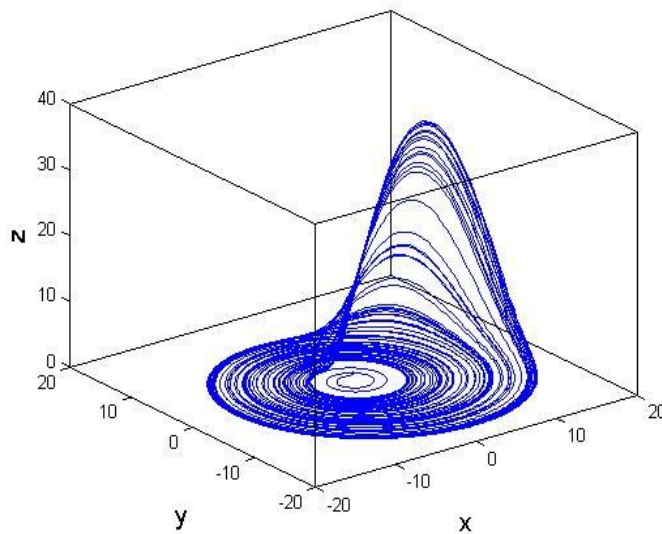


Fig. 4.24. The appearance of the Rössler attractor.

From this plot, this Rössler system generates a simpler chaotic attractor with a single lobe, compared with Lorenz attractor which has two lobes. The trajectory is generated by using the Runge-Kutta method with 0.001 second max step. The trajectory is integrated from an initial condition of (1, 1, 1).

For the Rössler attractor, the Average Mutual Information is shown in Fig. 4.25.

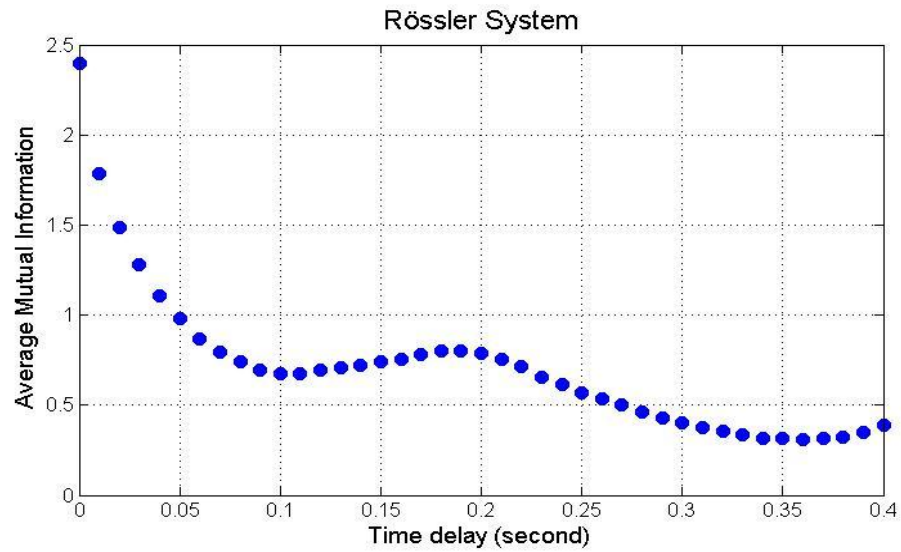


Fig. 4.25. Average mutual information of the Rössler attractor.

After the ideal time delay 0.1second is obtained, the False Nearest Neighbors method with 0.01 of the fraction R is applied to acquire the embedding dimension. The result is three dimensions for this Rössler system. The plot of the percentage of the false nearest neighbors versus the embedding dimension is shown in Fig. 4.26.

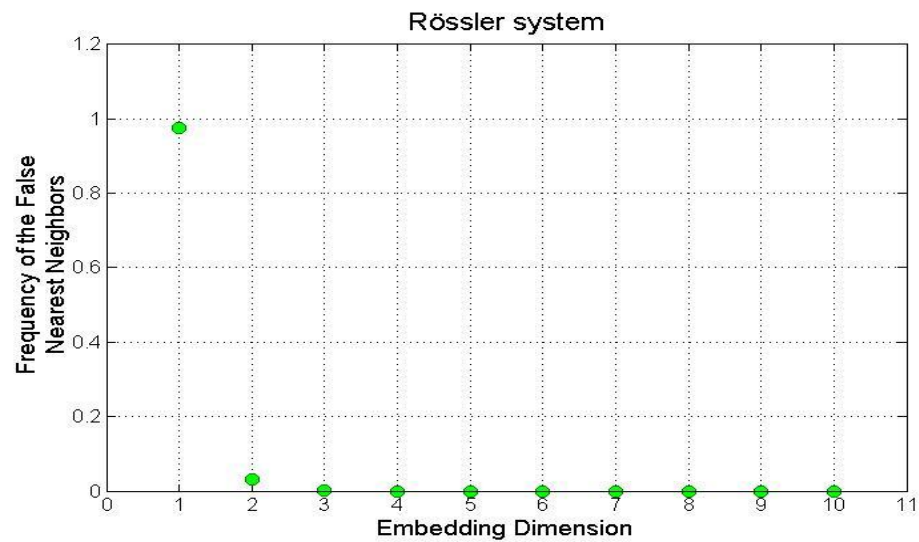


Fig. 4.26. The result of applying the False Nearest Neighbors method to the Rössler attractor.

The average value of ten evaluation times is 0.0914, which is close to the theoretical value 0.9.

4.5 Chapter Summary

The method for calculating the LLE is applied to the Lorenz attractor, the Hénon map, and the Rössler attractor. From these results, the estimated value matches the theoretical value very well. Though the calculated value has a little randomness because of the initial conditions of the weights and the biases of the neural network, the average value is usually accurate and the standard deviation is small. In addition, the neural network must be regularly updated because of the noise and the parameter variations. Based on the analysis of these theoretical models, the method has been proven to be feasible. In the next chapter, the method is applied to a real-world data set with the goal of finding a way to detect mechanical faults.

CHAPTER V

APPLICATION OF LARGEST LYAPUNOV EXPONENT TO FAULT DETECTION

5.1 Introduction

In this chapter, based on the algorithm of calculating the largest LLE, this metric is used to detect faults in dynamical systems. In the next section, the proposed method is introduced. Then, a real world system of an induction motor and the acquired data set from the stator currents are described. The proposed method is applied on this data set, and the results show some relationships between the damage levels and the percentage changes of the LLE. Finally, the summary of method and the conclusion of the application for fault detection are given.

5.2 Proposed Method for Incipient Fault Detection

There are varieties of faults which are hard to detect directly from the data itself. The LLE is the indicator of divergence or convergence of two trajectories with nearby initial conditions, and it could be sensitive to small changes of the systems. However, the relationship between the LLE and the system damage level is not clear and rarely studied up to now.

In the previous chapter, the calculated LLE has already been applied to three simple systems: Lorenz, Hénon and Rössler. However, the theoretical LLE is not known for real world data. From dynamic analysis of the Lorenz system, it is observed that the LLE varies following the changes in the system's parameters no matter whether the LLE

is positive or negative. In the real world, faults may cause changes in the parameters, and the parameters may change the LLE. Therefore, the changes of the LLE may exhibit the occurrence of the system's faults. Based on the LLE calculation method and considering the conditions of a real system and the measured data, the changes of the LLE are applied to fault detection.

5.3 A Real World Example

The time series experimental data was obtained from the stator currents I_a , I_b , I_c of an induction motor. In the experiment, the bearing of the motor was damaged in four steps. In each step, some currents were injected for some hours to put excessive stress on the bearing to compromise it. Then the current injection was stopped, and the motor was operated in that state for a number of days. Finally, the bearing was badly damaged. The steps used in the staged motor fault are shown in Fig. 5.1. The healthy state is named as damage level 0.

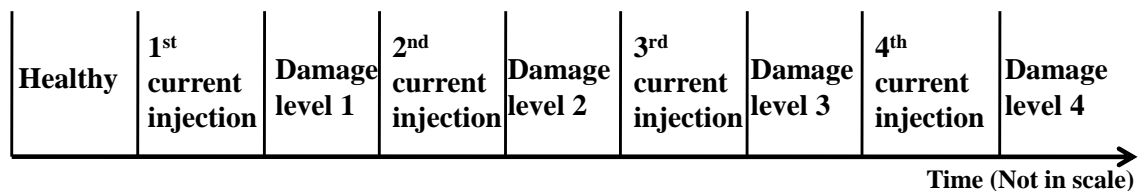


Fig. 5.1. Steps used in staged motor fault.

For each state, several data files are recorded. The sampling rate of the obtained data is 8 kHz, and each file contains 30 seconds of signals. The first available data file is file #10598 which corresponds to the healthy system, so the data from the file #10598 are considered as the baseline situation of the system.

In order to better understand the experimental data, two time series plots as the samples are shown in Fig. 5.2. The first sample is the data of the current I_a from the file #10598, and the second sample is the data of the current I_a from the file #57292 which belongs to the damage level 4.

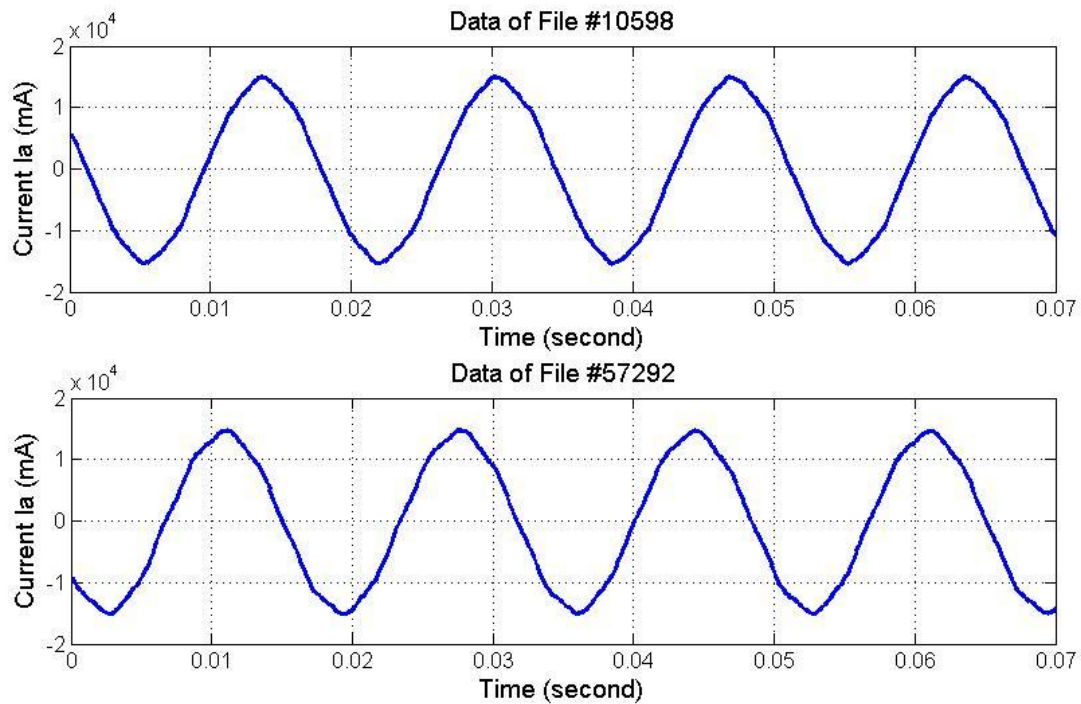


Fig. 5.2. The time series plots of the data from (a) data file #10598 (b) data file #57292.

From Fig. 5.2, it is difficult to distinguish the differences between the healthy situation and the situation of badly damaged bearings. Therefore, the method introduced in Chapter III is used to calculate the LLE for each file, and the calculation results and the analysis are displayed in the next section.

5.4 Experimental Results

For studying fault detection by using the LLE, the first step is to separate and select the data groups. The four current injection stages are ignored due to their unsteady behavior, and the five steady states are considered: damage level 0 to damage level 4. These 5 cases are used to evaluate their LLE and compare their changes.

The time delay and the embedding dimension are determined by the methods described in Chapter III. By using the data of the file #10598 with 8 kHz sample rate, the Average Mutual Information is obtained and its changes through different time delay are shown in Fig. 5.3.

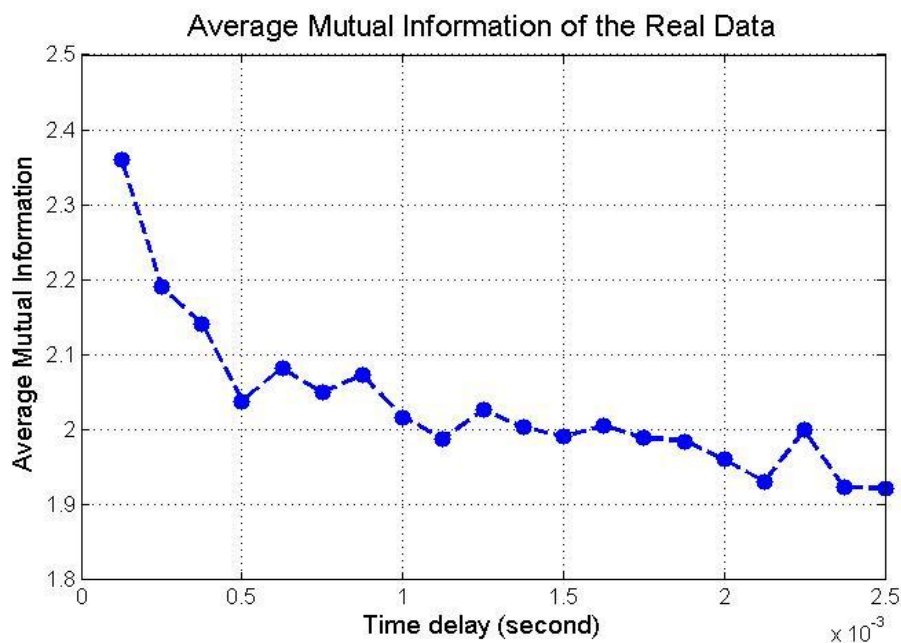


Fig. 5.3. Average mutual information of the data from the file #10598.

From Fig. 5.3, the first minimum value occurs at around 0.0005 seconds ($1/2000 \text{ Hz}^{-1}$). Since the frequency of the stator currents is 60 Hz, 1920 Hz (32×60) frequency is

usually considered as the resample frequency to reduce the background noise. Therefore, the time delay is set to equal to the sampling time (the inverse of the sampling rate).

In order to obtain high accuracy, the fraction R is set to be 0.001, and the false-nearest-neighbors method with this fraction R is applied to determine the embedding dimension. Then the plot of the percentage of the false nearest neighbors versus the embedding dimension is shown in Fig. 5.4. The left plot is full view, and the right plot is a zoomed in for clarity. The embedding dimension is set to be 4.

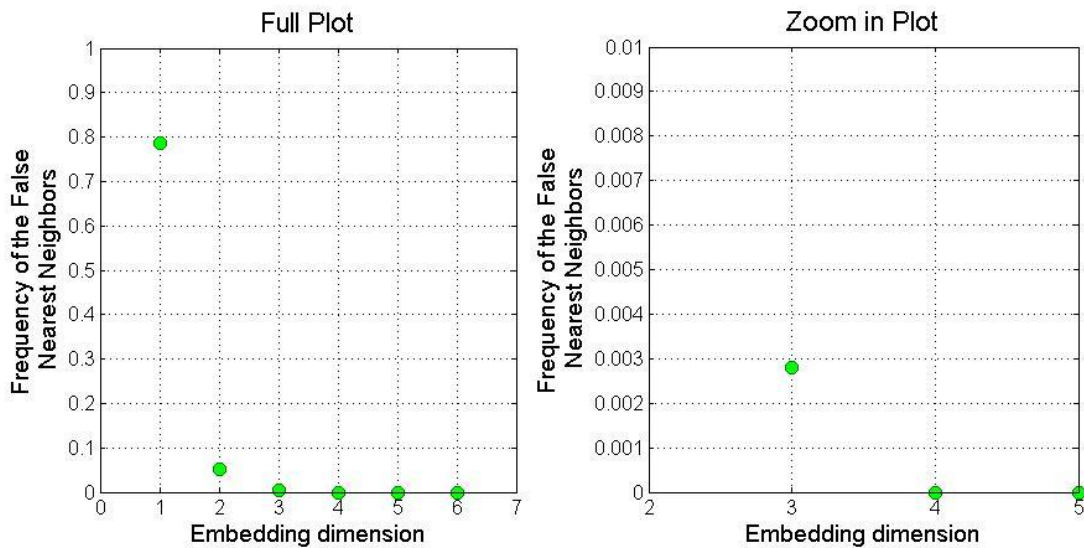


Fig. 5.4. The result of applying the False Nearest Neighbors method to the time series data from the induction motor system.

Based on the setup time delay and the embedding dimension, the data set is considered as the input for the multilayer neural network to train. After automatically training, the evaluation of the LLE can be obtained. The two files #10598 and #57292 are applied and compared. The comparison plots for the changes of the LLE through 20 seconds time length of the used data are shown in Fig. 5.5. The top plot is for the file #10598, and the bottom plot is for the file #57292.

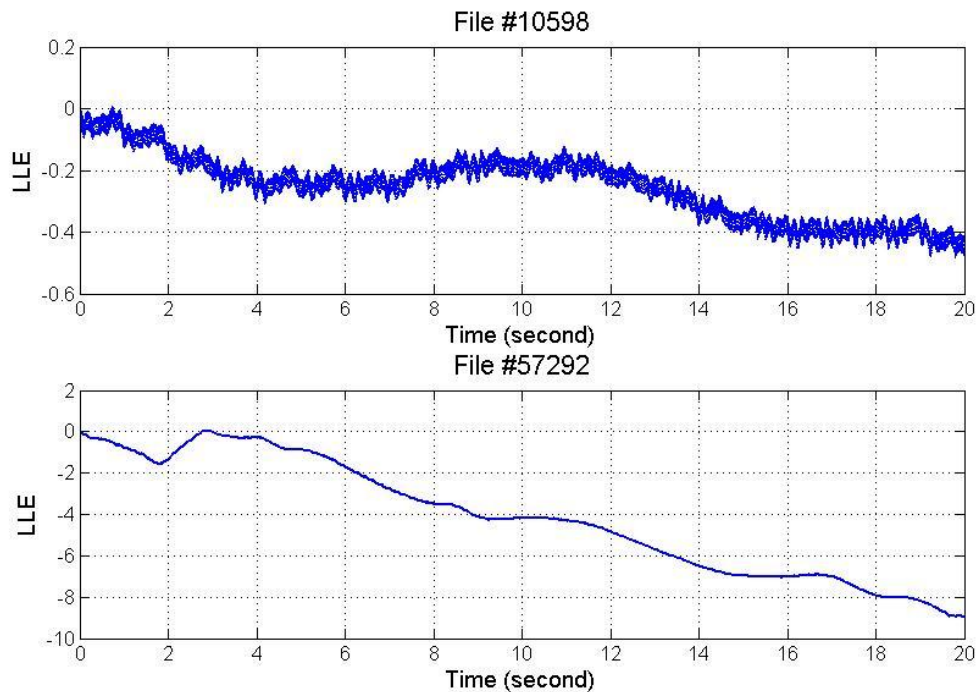


Fig. 5.5. Comparison of the Lyapunov exponents for the file #10598 and the file #57292 for 20 seconds of the evaluation time.

From the Fig. 5.5, the differences are easily found by the final values of the LLEs for these two files. The baseline file has the exponent of about -0.43, and -9 of the exponent is for the data file of the damaged bearing situation.

Two data files can not accurately represent the real situations. To further study the changes of the LLE and their distribution for different damage levels, more files are used to do the experiments. For the case of the damage level 0, 30 data files are used; for the damage level 1, only 7 files are used (because only 7 files are available); for the damage level 2 to 4, 50 data files are used, respectively.

The value of the LLE itself has less meaning since the theoretical value is unknown. Thus the percentage changes of the LLE for each file from the LLE of the baseline file are calculated as

$$\text{Percentage change} = \frac{\text{LLE} - \text{Baseline LLE}}{\text{Baseline LLE}} \times 100\% . \quad (5.1)$$

The percentage changes plot of the current Ia for all selected data files are displayed in Fig. 5.6. Similar results from currents Ib and Ic can be obtained as well.

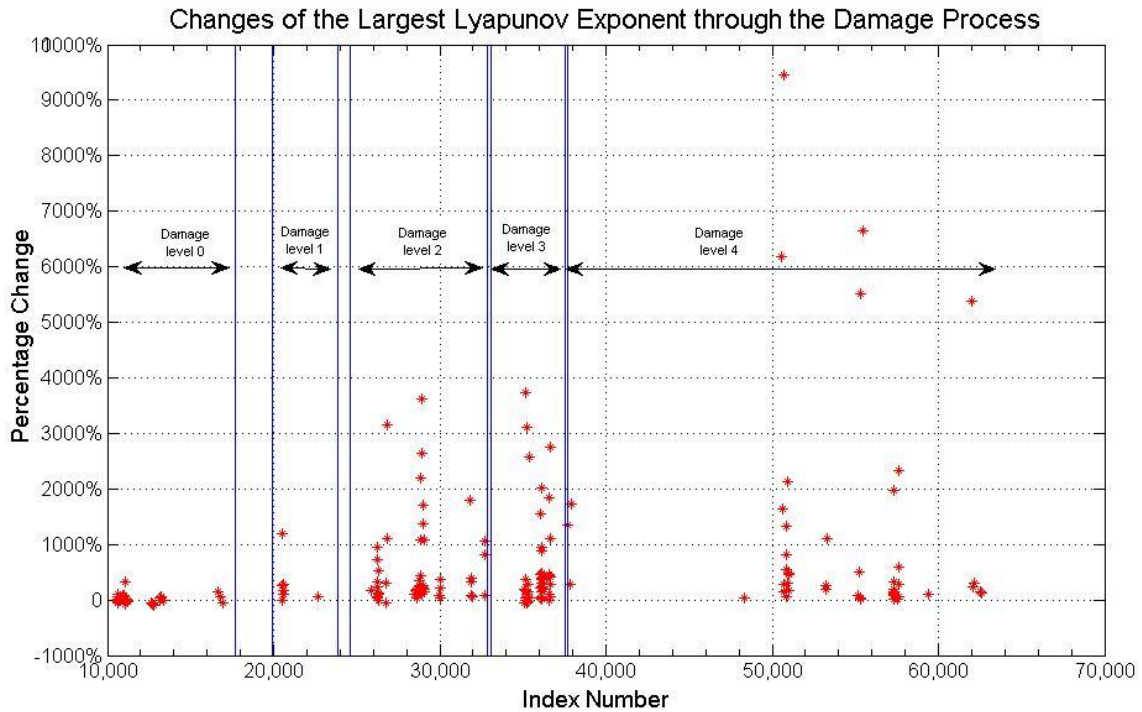


Fig. 5.6 Changes of the LLE for the current Ia through the entire damage process.

From Fig. 5.6, it is generally observed that the results of the percentage changes are not always the same for each damage level. Actually, the results are scattered, especially for the case of damage level 4. The extent of the separation tends to be larger as the damage level increases and the bearing becomes worse. In order to get more clear and convincing conclusion, the average value of percentage changes for each damage

level with the error bars of ± 1 standard deviations are presented in Fig. 5.7. The plot is for the current Ia, whereas the other two currents have the similar results.

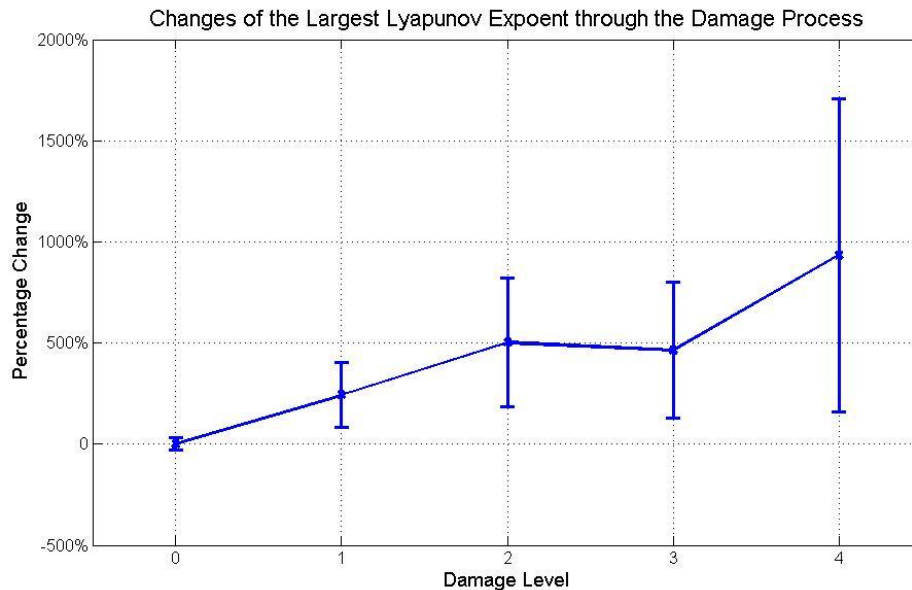


Fig. 5.7. The average value of the percentage changes of the LLE with error bars for each damage level.

Except for damage level 2 and damage level 3 which have similar values, the trend line is tending up as the damage level is increasing. Furthermore, the slope of the trend is related to the conditions before injecting currents and the current injection time. The 1st current injection is 25.38 hours; the 2nd current injection is 8 hours; the 3rd current injection is 2.47 hours; the 4th current injection is 1.38 hours. The slope from damage level 0 to damage level 1 and the slope from damage level 1 to damage level 2 are not very sharp, because before injecting current, the state is healthy or damaged very little though the current is injected for a lot of hours. The values for damage level 2 and damage level 3 are almost the same, since the time of injection current is short. Finally, the slope from damage level 3 to damage level 4 is sharp, because the bearing is close to

complete damage. So, only after a short time, approximately 1.38 hours, the bearing becomes badly damaged.

For additionally confirmation the trend with the damage process, the maximum of the percentage changes of the LLEs of the current Ia for each damage level are shown and compared in Fig. 5.8. The trend of this plot is similar to the plot for average values. The average, maximum, minimum and standard deviation of the LLE for each damage level for the three currents Ia, Ib, Ic are listed in Table B.1-B.3 in Appendix B.

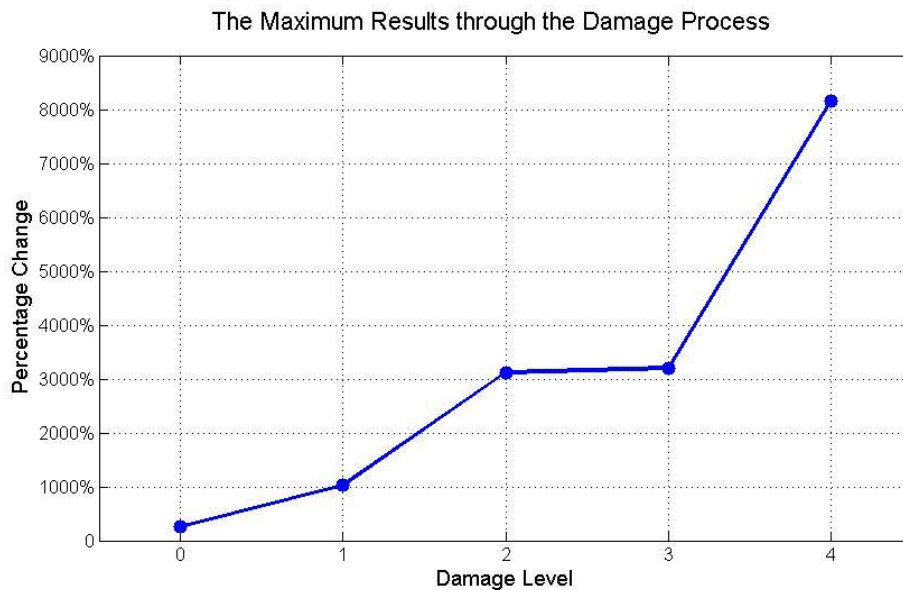


Fig. 5.8. The maximum value of the percentage changes of the LLE for each damage level.

Based on all of these experiments, the results are scattered no matter which case it is, but the distribution and the statistics analysis of results can indicate the changes both of the currents and the bearing conditions. In order to show clear results of the differences between the healthy system and the faulty system, the distribution frequency is shown for comparison in Fig. 5.9.

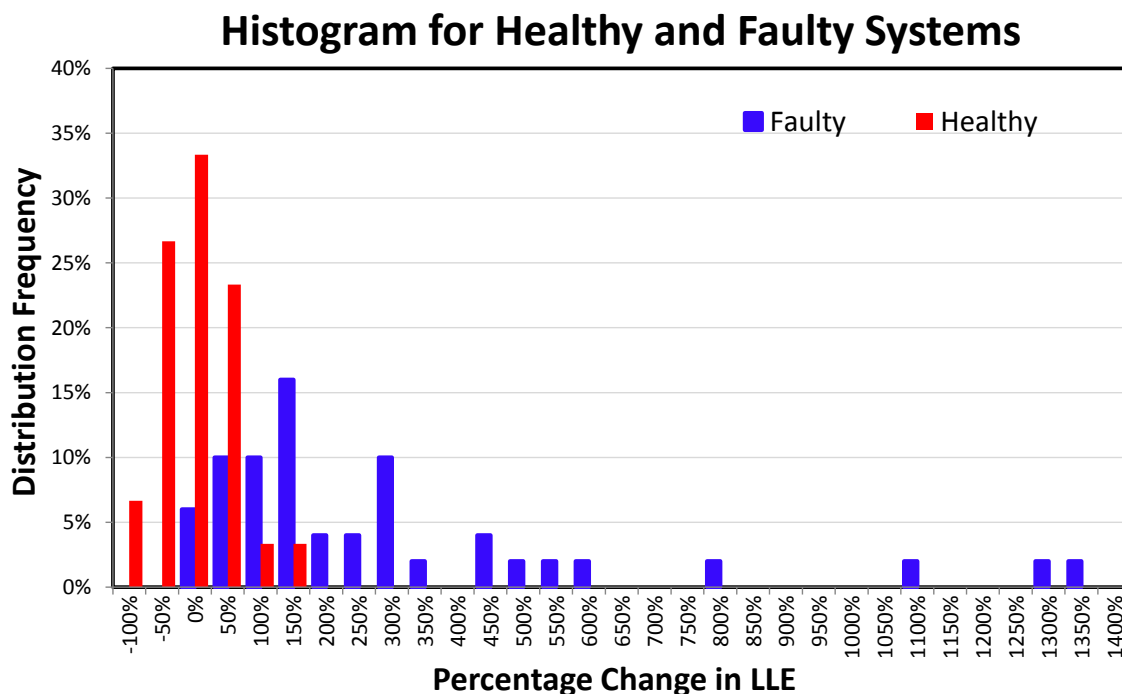


Fig. 5.9. Comparison of the distributions for the healthy system and the faulty system.

From the Fig. 5.9, the faulty system has a broadened histogram and several very large percentage changes in LLE compared with healthy system. The changes of the distribution can be a clear indicator for detecting faults for this real system.

5.5 Chapter Summary

From the results of testing the proposed method, a clear ascent trend of the percentage changes following the damage levels is shown. No matter which variable of the three currents is tested, the result is always similar. In addition, the distribution frequency of the percentage changes of the LLEs can give us a clear ability to distinguish and detect faults of a system.

CHAPTER VI

SUMMARY AND CONCLUSION

6.1 Summary

Chaos is a developing research topic, but its application in a number of scientific fields has a promising future. As the most important indicator of chaos, Lyapunov exponent (LE) is studied by lots of researchers. They developed several calculation methods for LE and applied them onto the studies of chaotic attractors. However, the application of Lyapunov exponent on detecting the faults in real world dynamical systems is rarely developed. So, the method of detecting faults by using the LLE is proposed in this thesis work.

In order to obtain the LLE from the experimental time series data, a method mainly based on phase space reconstruction and neural networks is demonstrated. The phase space reconstruction requires a time delay and an embedding dimension. For estimating the best time delay, the average mutual information is evaluated and the first minimum value is taken as the criterion. For evaluating the embedding dimension, the method of false nearest neighbors is used. Then the multilayer feedforward neural network is introduced to train the state points in the reconstructed phase space to predict the sub-functions of the unknown system and their derivatives. The last step is to calculate the eigenvalues of the Jacobi matrix through QR decomposition for simplification to attain the Lyapunov exponents.

Before applying the method on some real data, three chaotic models, the Lorenz, the Hénon and the Rössler, are used to test the method. After the analysis of the calculation results by using the data from these three theoretical models, the method is applied to some real world data from an induction motor during the damage bearing process. The relationship between the system fault and the calculated LLE is presented.

6.2 Conclusion

The conclusions of this thesis work are as following:

- The distribution of the LLE's percentage change can reflect changes in a mechanical system.
- The change of the LLE is a good indicator to detect the occurrence of faults.

6.3 Limitation and Future work

The proposed method for calculating the LLE from the time series can give a very good evaluation for the simple chaotic systems or the systems with a little noise. Nevertheless, when the signal-to-noise ratio is relatively small which means the amplitude of the noise is large, the calculated result cannot perfectly match the theoretical value. In addition, the trained neural network needs to be updated in time, because the parameters of the network are fixed after training. If the system has a subtle change, the evaluation based on the previous network will not be exactly correct. So, the robustness and feasibility of the method must be improved, and the performance of the method in the presence of noise is needed to be enhanced.

The result of the application of the method to the real world data shows the separation but not a specific value. In order to reduce this separation, more data is needed. However, the more data is used, the more costs of time and memories are needed. Thus, the ways of shortening the implementation time are needed to be explored in the future. Also, the accuracy of the calculated Lyapunov exponent from the real data cannot be proved since the theoretical value is not known. So, how to double check the calculated Lyapunov exponent is a significant future work for the application on fault detection. Lastly, other failure modes need to be further tested.

REFERENCES

- [1] J. Altmann and J. Mathew, "Multiple band-pass autoregressive demodulation for rolling element bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 963-977, Sep. 2001.
- [2] C. Junsheng, Yu Dejie and Y. Yu, "The application of energy operator demodulation approach based on EMD in machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 668-677, Feb. 2007.
- [3] H. Ocak, K. A. Loparo and F. M. Discenzo, "Online tracking of bearing wear using wavelet packet decomposition and probabilistic modeling: A method for bearing prognostics," *Journal of Sound and Vibration*, vol. 302, no. 951-961, May. 2007.
- [4] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, no. 2, pp. 130-141, Mar. 1963.
- [5] V. I. Oseledec, "A multiplicative ergodic theorem: Lyapunov characteristic numbers for dynamical systems," *Transactions of the Moscow Mathematical Society*, vol. 19, pp. 197-231, 1968.
- [6] A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285-317, Jul. 1985.
- [7] C. Rhodes, "False-Nearest Neighbors algorithm and noise-corrupted time series," *Physical Review Letters*, vol. 55, no. 5, pp. 6162-6170, May. 1997.

- [8] M. B. Kennel and H. D. I. Abarbanel, "False neighbors and false strands: A reliable minimum embedding dimension algorithm," *Physical Review Letters*, vol. 66, no. 2, pp. 1-18, 2002.
- [9] M. Lei, Z. Wang and Z. Feng, "A method of embedding dimension estimation based on symplectic geometry," *Physics Letters A*, vol. 303, no. 2-3, pp. 179-189, Oct. 2002.
- [10] M. T. Rosenstein, J. J. Collins and C. J. De Luca, "A practical method for calculating largest Lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, pp. 117-134, 1993.
- [11] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physics Review A*, vol. 33, no. 2, pp. 1134-1140, Feb. 1986.
- [12] N. J. I. Mars and G. W. van Arrangon, "Time delay estimation in non-linear systems using average amount of mutual information analysis," *Signal Processing*, vol. 4, no. 2-3, pp. 139-153, Apr. 1982.
- [13] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065-1076, Sep. 1962.
- [14] Y.-II Moon, B. Rajagopalan and U. Lall, "Estimation of mutual information using kernel density estimators," *Physics Review E*, vol. 52, no. 3, pp. 2318-2321, Sep. 1995.

- [15] H. Ma and C. Han, "Selection of embedding dimension and delay time in phase space reconstruction," *Frontiers of Electrical and Electronic Engineering in China*, vol. 1, no. 1, pp. 111-114, 2006.
- [16] M. Otani and A. J. Jones, "Automated embedding and creep phenomenon in chaotic time series," 2000 [Online]. Available:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.4473>
- [17] R. Gencay and W. D. Dechert, "An algorithm for the n Lyapunov exponents of an n-dimensional unknown dynamical system," *Physica D: Nonlinear Phenomena*, vol. 59, no. 1-3, pp. 142-157, Oct. 1992.
- [18] K. Gurney, *An Introduction to Neural Networks*. London: UCL Press, 1997.
- [19] S. Haykin, *Neural Networks a Comprehensive Foundation*, 2nd ed. Delhi: Pearson Education, pp. 121-220, 1994.
- [20] D. F. McCaffrey, S. Ellner, A. R. Gallant and D. W. Nychka, "Estimating the Lyapunov exponent of a chaotic system with nonparametric regression," *American Statistical Association*, vol. 87, no. 419, pp. 682-695, Sep. 1992.
- [21] M. Shintani and O. Linton, "Nonparametric neural network estimation of Lyapunov exponents and a direct test for chaos," *Journal of Econometrics*, vol. 120, no. 1, pp. 1-33, May 2004.
- [22] P. Xu, "Differential phase space reconstructed for chaotic time series," *Applied Mathematical Modelling*, vol. 33, no. 2, pp.999-1013, Feb. 2009.

- [23] G. A. Gottwald and I. Melbourne, "A new test for chaos in deterministic systems," *Proceeding of the Royal Society of London. Series A*, vol. 460, no. 2042, pp 603-611, Feb. 2003.
- [24] I. Falconer, G. A. Gottwald, I. Melbourne and K. Wormnes, "Application of the 0-1 test for chaos to experimental data," *Society for Industrial and Applied Mathematics*, vol. 6, no. 2, pp. 395-402, Jun. 2007.
- [25] G. A. Gottwald and I. Melbourne, "On the implementation of the 0-1 test for chaos," *Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 129-145, Jan. 2009.
- [26] B. D. Storey, "Computing Fourier series and power spectrum with MATLAB," 2002 [Online]. Available: <http://faculty.olin.edu/bstorey/Notes/Fourier.pdf>
- [27] H. P. F. Swinnerton-Dyer, "The calculation of power spectra," *The Computer Journal*, vol. 5, no. 1, pp. 16-23, 1962.
- [28] M C Valsakumar and S V M Satyanarayana, "Signature of chaos in power spectrum," *Journal of Physics*, vol. 48, no. 1, pp. 69-85, Jan. 1997.
- [29] P. C. Muller, J. Bajkowski and D. Soffker, "Chaotic motions and fault detection in a cracked rotor," *Nonlinear Dynamics*, vol. 5, no. 2, pp.233-254, 1994.
- [30] Zhen Zhao, Fu-Li Wang, Ming-Ming Jie, and Shu Wang, "Intermittent-chaos-and-cepstrum-analysis-based early fault detection on shuttle valve of hydraulic tube tester," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp.2764-2770, Jul. 2009.

- [31] P. T. A. Missert, "Chaos in a Dripping Faucet", [Online]. Available:
<http://www.pas.rochester.edu/~advlab/class2008>.
- [32] F. C. Moon, "Chaotic vibrations of a magnet near a superconductor," *Physics Letters A*, vol. 132, no. 5, Oct. 1988.
- [33] "Torus" Wikipedia. Available: <http://commons.wikimedia.org/wiki/File:Torus.png>.
- [34] J. C. Sprott, *Chaos and Time-Series Analysis*. Oxford: Oxford University Press, pp.116-117, 2003.
- [35] P. Gaspard, "Rossler systems," *Encyclopedia of Nonlinear Science*, New York: Routledge, pp. 808-811, 2005.

APPENDIX A

The calculation results of the average value, the relative error and the standard deviation of LLE for the Lorenz system with different time length are shown in Table A.1.

Table A.1. The LLEs for the Lorenz system with different time length.

	50	100	150	200	250	300	350	400
Average value	1.4166	0.9244	0.9350	0.9060	0.9095	0.9130	0.9259	0.9118
Relative %error	57.01%	2.46%	3.64%	0.42%	0.80%	1.20%	2.62%	1.06%
Standard deviation	0.2536	0.0794	0.0333	0.0340	0.0144	0.0270	0.0230	0.0266

The changed parameters of the Lorenz systems and the corresponding calculation results of the LLEs are listed in Tables A.2 - A.5.

Table A.2. The LLEs for the 5% and 10% parameters changed Lorenz systems without retraining the neural network.

CASE	0	1	2	3	4	5	6	7	8	9	10	11	12
SIGMA1	10	10	10	10	10	10	10	10	10	10	10	10	10
R1	28	28	28	28	28	28	28	28	28	28	28	28	28
BETA1	2.67	2.67	2.67	2.67	2.67	2.67	2.67	2.67	2.67	2.67	2.67	2.67	2.67
SIGMA2	10	10.5	9.5	10	10	10	10	11	9	10	10	10	10
R2	28	28	28	29.4	26.6	28	28	28	28	30.8	25.2	28	28
BETA2	2.67	2.67	2.67	2.67	2.67	2.8	2.5333	2.67	2.67	2.67	2.67	2.9333	2.4
Theoretical value	0.9022	0.9007	0.8997	0.9434	0.872	0.9202	0.8953	0.9018	0.9024	0.964	0.8344	0.9268	0.8704
Average value	0.9002	0.7053	1.0624	1.1591	0.7378	1.1688	0.7206	0.6202	1.1836	1.2833	0.9440	1.3794	0.7755
Relative %error	0.0022	0.2169	0.1808	0.2286	0.1539	0.2701	0.1952	0.3123	0.3116	0.3312	0.1313	0.4883	0.1090
Standard deviation	0.0143	0.0260	0.0258	0.0219	0.0246	0.0305	0.0344	0.0481	0.0312	0.0133	0.0734	0.0330	0.0922

Table A.3. The LLEs with the neural network trained by 3% parameters changed Lorenz system data.

CASE	1	2	3	4	5	6
SIGMA1	10.3	9.7	10	10	10	10
R1	28	28	28.84	27.16	28	28
BETA1	2.67	2.67	2.67	2.67	2.75	2.59
SIGMA2	10.5	9.5	10	10	10	10
R2	28	28	29.4	26.6	28	28
BETA2	2.67	2.67	2.67	2.67	2.80	2.53
Theoretical value	0.9007	0.8997	0.9434	0.872	0.9202	0.8953
Average value	0.8522	0.9807	1.0408	0.8052	1.0348	0.8137
Relative %error	0.0539	0.0900	0.1032	0.0766	0.1245	0.0912
Standard deviation	0.0200	0.0136	0.0108	0.0207	0.0198	0.0334

Table A.4. The LLEs with the neural network trained by 5% parameters changed Lorenz system data.

CASE	1	2	3	4	5	6
SIGMA1	10.5	9.5	10	10	10	10
R1	28	28	29.4	26.6	28	28
BETA1	2.67	2.67	2.67	2.67	2.80	2.53
SIGMA2	10.5	9.5	10	10	10	10
R2	28	28	29.4	26.6	28	28
BETA2	2.67	2.67	2.67	2.67	2.80	2.53
Theoretical value	0.9007	0.8997	0.9434	0.872	0.9202	0.8953
Average value	0.9005	0.8972	0.9397	0.8862	0.9139	0.8988
Relative %error	0.0002	0.0028	0.0040	0.0162	0.0069	0.0039
Standard deviation	0.0257	0.0188	0.0207	0.0119	0.0215	0.0248

Table A.5. The LLEs with the neural network trained by 10% parameters changed Lorenz system data.

CASE	1	2	3	4	5	6
SIGMA1	11	9	10	10	10	10
R1	28	28	30.8	25.2	28	28
BETA1	2.67	2.67	2.67	2.67	2.93	2.40
SIGMA2	11	9	10	10	10	10
R2	28	28	30.8	25.2	28	28
BETA2	2.67	2.67	2.67	2.67	2.93	2.40
Theoretical value	0.9018	0.9024	0.964	0.8344	0.9268	0.8704
Average value	0.9064	0.9255	0.9720	0.8361	0.9323	0.8917
Relative %error	0.0051	0.0256	0.0082	0.0020	0.0060	0.0244
Standard deviation	0.0191	0.0196	0.0192	0.0174	0.0257	0.0169

The Sinusoidal functions with different frequencies and different amplitude are added to the Lorenz system. The calculation results of the LLEs are shown in Tables A.6

- A.8.

Table A.6. The largest Lyapunov exponent of the Lorenz system added Sinusoidal function input with the amplitude of 10.

Frequency(Hz)	1	2	3	4	5	6	7	8	9	10
Theoretical value	-0.4244	-0.5406	-1.0450	0.1312	0.5746	1.0586	0.7556	0.7668	0.8178	0.8650
Average value	-7.9764	-8.5523	-4.3038	-9.1729	-0.6281	1.6826	1.5324	1.2572	1.1042	1.1416
Relative %error	1779.34%	1481.90%	311.85%	7093.00%	209.31%	58.94%	102.81%	63.95%	35.02%	31.98%
Standard deviation	3.4817	2.9762	1.2410	1.8475	0.5945	0.5376	0.3617	0.1461	0.0935	0.1392

Table A.7. The largest Lyapunov exponent of the Lorenz system added Sinusoidal function input with the amplitude of 5.

Frequency(Hz)	1	2	3	4	5	6	7	8	9	10
Theoretical value	-0.6512	-0.1443	0.7724	0.9018	1.0375	0.7746	0.8187	0.8316	0.8539	0.8392
Average value	-4.4922	-6.4287	1.2937	1.8242	1.4101	1.4077	1.4870	1.1254	1.1431	1.0807
Relative %error	589.87%	4356.52%	67.49%	102.29%	35.92%	81.72%	81.62%	35.33%	33.87%	28.77%
Standard Deviation	1.3516	1.8251	0.4140	0.2437	0.1287	0.2204	0.2093	0.0966	0.0924	0.1013

Table A.8. The largest Lyapunov exponent of the Lorenz system added Sinusoidal function input with the amplitude of 1.

Frequency(Hz)	1	2	3	4	5	6	7	8	9	10
Theoretical value	1.03565	0.86118	1.03517	0.88218	0.79777	0.86022	0.84734	0.87619	0.87215	0.85614
Average value	2.59102	1.74142	4.25304	3.94714	1.78857	1.47720	1.02880	0.97005	1.00244	0.96445
Relative %error	150.18%	102.21%	310.85%	347.43%	124.20%	71.72%	21.42%	10.71%	14.94%	12.65%
Standard Deviation	0.27127	0.23720	0.81897	0.45499	0.17281	0.14867	0.13076	0.07198	0.09990	0.11754

The Gaussian white noise with different signal-to-noise ratio is added to the Lorenz system. The calculation results of the LLEs are shown in Tables A.9 - A.10.

Table A.9. The largest Lyapunov exponents evaluated from noise-added system data with the neural network trained by non-noise Lorenz system data.

SNR	4.7712			6.9897			10			13.0103			20		
Interval time	0	4000	8000	0	4000	8000	0	4000	8000	0	4000	8000	0	4000	8000
Average value	1.3368	1.3467	1.2238	1.2612	1.1476	1.2822	1.0674	1.0001	1.0048	0.9783	1.0321	0.9709	0.8905	0.9146	0.8839
Relative %error	48.17%	49.27%	35.65%	39.79%	27.20%	42.12%	18.31%	10.85%	11.37%	8.43%	14.39%	7.61%	1.30%	1.37%	2.03%
Standard deviation	0.0900	0.0792	0.0748	0.0680	0.0586	0.0655	0.0472	0.0456	0.0480	0.0286	0.0263	0.0332	0.0198	0.0214	0.0184

Table A.10. The largest Lyapunov exponents evaluated from noise-added system data with retraining neural network.

SNR	4.7712			6.9897			10			13.0103			20		
Interval time	0	4000	8000	0	4000	8000	0	4000	8000	0	4000	8000	0	4000	8000
Average value	2.4683	2.7334	2.5597	2.2221	2.0569	2.2068	2.1943	1.9878	1.8169	1.7660	1.8057	1.8190	1.4597	1.4693	1.4713
Relative %error	173.58%	202.97%	183.72%	146.29%	127.98%	144.60%	143.21%	120.32%	101.39%	95.75%	100.15%	101.61%	61.79%	62.86%	63.08%
Standard deviation	0.4844	0.5476	0.5156	0.6496	0.6798	0.7229	0.3304	0.1989	0.2678	0.3226	0.3766	0.3354	0.1373	0.1647	0.1124

APPENDIX B

The LLEs for the three currents of the motor, I_a , I_b and I_c , during the entire damage process are calculated. The average, maximum, minimum and standard deviation of the LLE for each damage level are listed in Tables B.1 - B.3.

Table B.1 The largest Lyapunov exponents of current I_a for 5 different damage levels.

Damage level	0	1	2	3	4
Average of the largest Lyapunov exponents	-0.44523	-1.52785	-2.68449	-2.5113	-4.60214
Maximum of the largest Lyapunov exponents	-0.0481	-0.37123	-0.1663	-0.208	-0.4059
Minimum of the largest Lyapunov exponents	-1.5969	-4.99735	-14.3552	-14.7197	-36.7801
Standard deviation	0.31196	1.584841	3.192624	3.374332	7.742202

Table B.2 The largest Lyapunov exponents of current I_b for 5 different damage levels.

Damage level	0	1	2	3	4
Average of the largest Lyapunov exponents	-0.54598	-1.07261	-2.78715	-2.35668	-4.6279
Maximum of the largest Lyapunov exponents	-0.0939	-0.3744	0.0142	-0.168	-0.1938
Minimum of the largest Lyapunov exponents	-1.7373	-3.05738	-14.0942	-15.7505	-39.9594
Standard deviation	0.330599	0.94629	3.37928	3.4081	8.314678

Table B.3 The largest Lyapunov exponents of current I_c for 5 different damage levels.

Damage level	0	1	2	3	4
Average of the largest Lyapunov exponents	-0.71333	-1.20474	-2.6908	-2.2429	-4.6698
Maximum of the largest Lyapunov exponents	-0.2008	-0.5014	-0.27485	-0.0798	-0.3985
Minimum of the largest Lyapunov exponents	-2.0737	-3.33783	-11.9814	-13.5551	-34.5237
Standard deviation	0.456921	0.987945	3.094062	3.108331	8.210244

VITA

Yifu Sun grew up in Beijing, China. He finished his B.S. degree in the Department of Automation Control at Beijing Institute of Technology, Beijing, China in 2008. He received his M.S. degree in the Department of Mechanical Engineering at Texas A&M University, College Station, Texas in 2011.

Contact Address: Texas A&M University
Department of Mechanical Engineering
3123 TAMU
College Station TX 77843-3123

Email Address: tom19851031@neo.tamu.edu

Phone Number: 979-739-3227