5-16-2008

Computational Intelligence and Complexity Measures for Chaotic Information Processing

Davoud Arasteh
University of New Orleans

COMPUTATIONAL INTELLIGENCE AND COMPLEXITY MEASURES FOR CHAOTIC INFORMATION PROCESSING

A Dissertation

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Engineering and Applied Science

by

Davoud Arasteh

B.S. (National University of Iran, Tehran) 1991
M.S. (Amir Kabir University of Technology, Tehran) 1994
M.S. (University of Louisiana, Lafayette) 2002

University of New Orleans
May 2008

DEDICATION

To my wife, Mehrdokht
my daughters,
Ailar and Aileen
my son, Aidin
for their endless love, support
and inspiration.

ACKNOWLEDGEMENTS

I am deeply indebted to my advisors Dr. George Ioup and Dr. Juliette Ioup for their continuous support and encouragement throughout my doctoral studies. Their patience, enthusiasm, and generous contribution of time and resources have greatly helped me to achieve my goals and kept this research progressing with good insight and advice. I am especially thankful for their trust in my ideas and abilities and for the opportunities they have offered me. I would like to thank the committee co-chair Dr. Abolfazl Amini for all the knowledge and support. He served on both my general examination and dissertation committees, encouraging me every step of the way. I am fortunate to have had the guidance and support of many faculties at UNO, especially Dr. Jinke Tang and Dr. Dimitrios Charalampidis. They generously served on both general examination and dissertation committees.

I would like to thank the department staff especially Sandra Merz and Zella Huaracha. They are both very knowledgeable and genuinely friendly. They took care of every administrative bump I ran into.

Finally, I wish to thank and express my appreciation to my family for their immeasurable emotional support and encouragement, which was indispensable during my time in graduate school. Thank you for keeping the faith and getting me here.

During the years the material in this dissertation has been presented at conferences and parts have also been accepted for publication. Here are some of the places where it occurs in proceedings and journals.

- *Algorithmic Complexity Measure and Lyapunov Matrices of the Dynamical Systems*, Paper Number: ENT 108-099, 2006 IJME/Intertech International Conference Proceedings.
- *Algorithmic Complexity Measure of the Dynamical Systems*, Intelligent Engineering Systems through Artificial Neural Networks, Volume 16, pp. 161-167, ASME Press 2006.
- *Ultra High-Speed Microbridge Chaos Domain*, Intelligent Engineering Systems through Artificial Neural Networks, Volume 17, pp. 365-370, ASME Press 2007.
- *Computing Algorithmic Complexity Using Advance Sampling Technique*, Intelligent Engineering Systems through Artificial Neural Networks, to appear in Volume 18, ASME Press 2008.
- *Algorithmic Complexity Analysis of Best ½-Rate Convolutional Codes with Chaotic-Encrypted Data*, Intelligent Engineering Systems through Artificial Neural Networks, to appear in Volume 18, ASME Press 2008.
- *Measures of Order in Dynamic Systems*, Paper Number: CND-06-1119, Journal of Computational and Nonlinear Dynamics, to appear in Volume 3, July 2008.
- *Rate-1/n Maximum Free Distance Convolutional Codes Complexity with Chaotic-Encrypted Data*, to be submitted to IEEE Computational Intelligence Magazine, 2008.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ATS	Advance-Time Sampling
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CNM	Convolutionally Coded Not Masked
CRC	Cyclic Redundancy Code
DF	Duffing oscillator
FEC	Forward Error Correcting
GSM	Global System for Mobile Communication
GSR	Gram-Schmidt Reorthonormalization
HiperLAN-2	High-performance Local Area Network type 2
LE	Lyapunov Exponent
LZ	Lempel-Ziv
LZC	Lempel-Ziv Complexity
LZC_N	Normalized Lempel-Ziv Complexity
ΔLZC	Lempel-Ziv Complexity increment
ΔLZC_{AVE}	Average Lempel-Ziv Complexity increment
MC	Masked and Convolutionally Coded
MPEG	Moving Picture Expert Group
ML	Maximum Likelihood
MNC	Masked No Coding
NRNO	No Rescaling No Orthogonalization
OVSF	Orthogonal Variable Spreading Factor
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RCSJ	Resistively and Capacitively Shunted Junction
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SQUID	Superconducting Quantum Interference Device
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
VP	Van der Pol oscillator
VSAT	Very Small Aperture Terminal
WCDMA	Wideband Code Division Multiple Access

ABSTRACT

This dissertation investigates the application of computational intelligence methods in the analysis of nonlinear chaotic systems in the framework of many known and newly designed complex systems. Parallel comparisons are made between these methods. This provides insight into the difficult challenges facing nonlinear systems characterization and aids in developing a generalized algorithm in computing algorithmic complexity measures, Lyapunov exponents, information dimension and topological entropy. These metrics are implemented to characterize the dynamic patterns of discrete and continuous systems. These metrics make it possible to distinguish order from disorder in these systems. Steps required for computing Lyapunov exponents with a reorthonormalization method and a group theory approach are formalized. Procedures for implementing computational algorithms are designed and numerical results for each system are presented.

The advance-time sampling technique is designed to overcome the scarcity of phase space samples and the buffer overflow problem in algorithmic complexity measure estimation in slow dynamics feedback-controlled systems.

It is proved analytically and tested numerically that for a quasiperiodic system like a Fibonacci map, complexity grows logarithmically with the evolutionary length of the data block. It is concluded that a normalized algorithmic complexity measure can be used as a system classifier. This quantity turns out to be one for random sequences and a non-zero value less than one for chaotic sequences. For periodic and quasi-periodic responses, as data strings grow their normalized complexity approaches zero, while a faster deceasing rate is observed for periodic responses.

Algorithmic complexity analysis is performed on a class of certain rate convolutional encoders. The degree of diffusion in random-like patterns is measured. Simulation evidence indicates that algorithmic complexity associated with a particular class of 1/*n*-rate code increases with the increase of the encoder constraint length. This occurs in parallel with the increase of error correcting capacity of the decoder. Comparing groups of rate-1/*n* convolutional encoders, it is observed that as the encoder rate decreases from 1/2 to 1/7, the encoded data sequence manifests smaller algorithmic complexity with a larger free distance value.

Keywords

Advance-time sampling technique, Chaos synchronization, Chaotic encryption, Convolutional coding, Cryptosystem algorithmic complexity, Feedback controlled systems, Lyapunov exponent spectrum, Lempel-Ziv algorithmic complexity, Secure communication.

CHAPTER 1

INTRODUCTION

1.1 Dissertation Contributions

The goal of our work is to develop practical methodology in evaluating the performance of nonlinear systems with complex models. Our work focuses on demonstrating and testing good performing metrics like the Lyapunov exponent spectrum, an algorithmic complexity measure, the fractal information dimension and entropy. We are motivated by the fact that the theoretical work tends to be too specialized to be useful for real deployments, and the practical implementations have yet to address fundamental practical challenges. Our work achieves its goals by implementing and evaluating these metrics in the framework of many known and newly designed complex systems like the tetrode microbridge with driving source voltage, coupled non-homogeneous synchronized oscillators, a class of feedback-controlled systems and a Fibonacci quasiperiodic system. Parallel comparisons are made between these techniques. This provides insight into the difficult challenges facing nonlinear systems characterization and aids in developing a generalized algorithm in computing Lempel-Ziv complexity (LZC). This work leads us in developing an advance sampling algorithm and a corresponding simulation to overcome the buffer overflow problem in feedback controlled systems. Focusing on these systems, we found that there is a similarity in the behavior of feedback controlled systems in the hyperchaos region which leads to attractor collapse and the disappearance of the dynamic state. It is found that LZC values for four-dimensional feedback controlled hyperchaotic system have a non-decreasing characteristic behavior just before the occurrence of attractor collapse when a

diverging dynamic state occurs. In chaotic regions, the LZC measure increases from low values to high, then drops in periodic windows, but does not sustain a non-decreasing envelop.

Our work extends the application of Lempel-Ziv complexity concept to the study of a quasiperiodic system and shows that for a system like the Fibonacci map, complexity grows logarithmically with the evolutionary length of the data block and proves the characteristics of the correlation factor in this system. It concludes that a normalized algorithmic complexity measure can be used as a general system classifier. Our final framework allows for extensions to chaotic cryptosystem design with convolutional encoders where an algorithmic complexity measure is used to evaluate the performance of convolutional encoders along with their error correcting capability.

1.2 Dissertation Outline

Our work is presented in the following manner. Chapter 2 reviews two methods for computing Lyapunov exponents and develops a practical approach with two methods in computing a Lyapunov exponent spectrum. In group theory approach the minimal number of variables is used, rescaling and reorthogonalization are eliminated, and a partial Lyapunov spectrum is calculated using a smaller number of equations. In low dimensional systems the modified Gram-Schmidt Reorthonormalization method converges quickly and the exponent spectrum appears in order starting from largest exponent. Also partial Lyapunov spectrum is calculated. Our experience with group theory method shows that calculated exponents mix and appear out of order. Numerical implementation of both methods in a Josephson junction and the Lorenz systems is provided in chapter 3 where main challenge categories along with potential solutions are discussed. This chapter concludes with a framework that unifies these solutions. Chapter 4 develops and uses an algorithmic complexity measure and compares it to the

Lyapunov exponent as a metric to characterize the dynamic patterns of discrete and continuous systems with chaotic, periodic, and quasi-periodic states. This is done by mapping the system output signal into a binary string and then performing the complexity measure analysis on the binary time-sequence.

In Chapter 5, we study the Josephson tetrode microbridge device in which a four-terminal superconductive device is made of five Josephson weak-link junctions through a microbridge configuration. A set of system models is developed for autonomous and non-autonomous systems in which two junctions are connected in series and three junctions are connected in parallel with three independent variables to satisfy the necessary condition for the generation of chaos. We use a resistively shunted Josephson junction model for numerical analysis. Several simulations are provided to demonstrate the ability of microbridge systems in pseudorandom noise generation and chaotic oscillations. We also calculate the Lyapunov exponent using the dynamics of electrical voltages across the junctions when one of the normal resistances is varied.

In chapter 6, we investigate complexity measure performance in hyperchaotic systems. For slow dynamics systems an advance sampling method is designed and implemented algorithmically to overcome the scarcity of phase space samples in algorithmic complexity measure estimation in feedback controlled hyperchaotic systems. Simulations are provided as evidence that this technique reduces complexity estimation error. Special attention is paid to the complexity of computing information for an entire region of chaos-hyperchaos transition.

In chapter 7, a non-homogeneous system of coupled nonlinear oscillators is designed. Two dynamic phase diagrams are constructed in terms of control parameters. This is done by computing the Lyapunov exponent spectrum and selecting the positive values as the indicator of chaos and order. Periodic regions are determined by bifurcation diagrams. A synchronization

study confirms that the coupled non-homogeneous chaotic oscillator system exhibits three forms of synchronization. That is, phase, complete and lag synchronization. Our simulation result predicts the occurrence of complete synchronization despite the fact that coupled oscillators are non-homogeneous.

In Chapter 8, we investigate the algorithmic complexity measure idea more formally, and compare the accuracy of performance of this measure to the Lyapunov exponent in an analytical framework in a quasi-periodic system. Our results prove analytically and numerically that for a system like the Fibonacci map, complexity grows logarithmically with the evolutionary length of the data block. We conclude that a normalized algorithmic complexity measure can be used as a system classifier. This quantity turns out to be one for random sequences and a non-zero value less than one for chaotic sequences. For periodic and quasi-periodic responses, as the data string grows their normalized complexity approaches zero, while a higher deceasing rate is observed for periodic responses.

Chapter 9 presents the algorithmic complexity analysis performed on a convolutionally encoded chaotic-encrypted message for an entire class of rate-1/n convolutional encoders. We design simulating programs that evaluate the efficiency and performance of this mechanism. Numerical evidence indicates that the algorithmic complexity associated with particular 1/n-rate convolutional encoders increases as constraint length increases. This occurs in parallel with the increase of error correcting capacity of the decoder or free distance. We conclude that algorithmic complexity is a suitable measure of the quality factor along with other measures of the weight structure of the code. Comparing groups of rate-1/n convolutional encoders, we observe that as the number of output streams n increases from 2 o 7 (1/n decreases from 1/2 to

1/7 rate) the encoded data sequence manifest a lower algorithmic complexity and higher free distance.

1.3 Introduction to Chaos

Dynamical systems are mathematical models that can be studied without reference to nature. Their applicability to real world phenomena has stimulated the interest in them. There are numerous chaotic systems in nature. For example normal brain activity is chaotic, and pathological order is indeed the cause of diseases such as epilepsy, or too much periodicity in heart rates might indicate disease [1, 2, 3, 4, 5]. In neural systems the divergence from the initial state is the fundamental characteristic of perception which distinguishes very close perceptual entities. Signals obtained from processes like EEG and ECG or behavioral signals appear to be random. Despite that, these signals are not random and can be classified as chaotic. Artificial cognitive informatics is concerned with the extraction of characteristic features from these signals, for use in measurement and characterization of patterns in the processes related to perception and cognition. In communication systems, it is shown that two identical synchronized sequences of chaotic signals can be used for encryption by superposing a message on one sequence [6]. Only a person with the other sequence can decode the message by subtracting the chaotic masking component. Recent development in nonlinear dynamics and chaos has led to the idea of realizing digital communication by utilizing devices operating in nonlinear regimes where a chaotic system can be manipulated, via arbitrarily small time-dependent perturbations, to generate controlled chaotic orbits whose symbolic representation corresponds to the digital representation of a desirable message or the information one wishes to encode [7, 8, 9, 10]. It is speculated that such control and sequences may be a possible mechanism by which biological systems transmit information. In computer networks efficient algorithms are developed for

secure packet communications in packet switching networks over the Internet Protocol (IP) using a Multilayer Chaotic Encryption (MCE) algorithm, or on top of a non-guarantee protocol like the Real-time Transport Protocol (RTP) using a chaotic real-time cryptosystem [11, 12].

Chaotic fluctuations can be used to stimulate trapped solutions in neural network problems so they escape from local minima for optimization or learning problems. Neural networks are modeled on biological neural networks, or the human brain, and learn by themselves from patterns. This learning can then be applied to control, predict and classify systems, such as speech production, speech and handwriting recognition and motor control [13]. Here chaotic behavior provides a rich library of behaviors to aid such adaptation. This characteristic also has been used in network architectures and learning algorithms by exploiting chaos and chaotic circuits for associative memory storage of analog patterns. Architectural variations employ selective synchronization of modules with chaotic behaviors that communicate through broad spectrum chaotic signals [14].

1.4 Basic concepts in dynamical systems

When we study dynamical systems, we study the evolution of quantities called state variables in time and space. To do this, we formulate governing equations. The strategy of modeling a dynamical system begins with the choice of a state space in which observations can be represented by several parameters. These collective observations lead to many trajectories within the state space. A velocity vector field is defined by prescribing a velocity vector at each point in the state space. The state space filled with trajectories is called the phase portrait of the dynamical system. This basic concept of dynamical system theory was originally introduced by Henri Poincare. The velocity vector field is derived from the phase portrait by differentiation. The velocity vector field visualizes the dynamics of the particular system being modeled.

Actually, extensive observations over a long period of time are necessary to reveal the dynamical tendencies of the system which is represented by the corresponding velocity vector field. The modeling procedure is only adequate if we assume that (a) a velocity vector of an observed trajectory is at each point exactly the same as the vector specified by the dynamical system and (b) the vector field of the model is smooth. The dynamical model consists of the state space and a vector field. The state space is a geometrical space like the Euclidean plane or in general a topological manifold of the experimental situation. The vector field represents the habitual tendencies of the changing states and is called the dynamics of the model. Given a state space and a smooth vector field, a curve in the state space is a trajectory of the dynamical system if its velocity vector agrees with the vector field in the sense of tangent vectors. These trajectories are supposed to describe the behavior of systems as observed over an interval of time. The point corresponding to time zero is called the initial state of the trajectory. The limit sets are the asymptotic destination of trajectories. In the case of a limit point, an attractor represents a static equilibrium, while a limit cycle as attractor designates the periodic equilibrium of an oscillation. In a typical phase portrait, there will be more than one attractor. The phase portrait will be divided into its different regions of approaching trajectories. The dividing boundaries or regions are called separatrices. For example, the motion of the pendulum will cease because of friction. The equilibrium stationary point is a limit-point for this system. This means any trajectory representing a slow motion of the pendulum near the bottom, approaches this limit point asymptotically. In two dimensions or more, other types of trajectories and limit sets may occur. For example, a cycle may be the asymptotic limit set for a trajectory. In a three-dimensional system a torus or even other more or less strange limit sets may occur.

In chaotic dynamics, the key concepts are limit sets called attractors. Mathematically, a limit set (limit point, cycle, torus, etc.) is called an attractor if the set of all trajectories approaching this limit set asymptotically is open. That is to say, attractors receive most of the trajectories in the neighborhood of the limit set. Of all limit sets which represent possible dynamical equilibrium of the system, the attractors are the most prominent. Limit sets enable us to model a system's evolution to its equilibrium states.

A dynamical system cannot be considered as isolated from other dynamical systems. An empirical application is given in the three-body problem of celestial mechanics, which is non-integrable. Consider the motion of Jupiter perturbing the motion of an asteroid around the sun. Jupiter and the asteroid are interpreted as two oscillators with certain frequencies. The observed chaotic behavior is neither due to a large number of degrees of freedom nor to the uncertainty of human knowledge. The irregularity is caused by the nonlinearity of the Hamiltonian equations which let initially close trajectories separate exponentially fast in a bounded region of phase. As their initial conditions can only be measured with finite accuracy, and errors increase exponentially fast, the long-term behavior of these systems cannot be predicted.

Conservative as well as dissipative systems are characterized by nonlinear differential equations with a nonlinear function of the state vector depending on an external control parameter. While for conservative systems, according to Liouville's theorem, the volume elements in the corresponding phase space change their shape but retain their volume in the course of time, the volume elements of dissipative systems shrink as time increases. The first dissipative system analyzed by computer-assisted simulation was the Lorenz model. Lorenz's discovery of a deterministic model of turbulence occurred during simulation of global weather patterns. The earth, warmed by the sun, heats the atmosphere from below. Outer space, which is

always cold, absorbs heat from the outer shell of the atmosphere. The lower layer of air tries to rise, while the upper layer tries to drop. This traffic of layers was modeled in several experiments by Bernard. The air currents in the atmosphere can be visualized as cross-sections of the layers. The traffic of the competing warm and cold air masses is represented by circulation vortices, called Bernard cells. In three dimensions, a vortex may have warm air rising in a ring, and cold air descending in the center. Thus, based on this model the atmosphere consists of a sea of three-dimensional Bernard-cells, closely packed as a vortex lattice of width h/a. A footprint of such a sea of atmospheric vortices can be observed in the regular patterns of hills and valleys in deserts, snowfields, or icebergs.

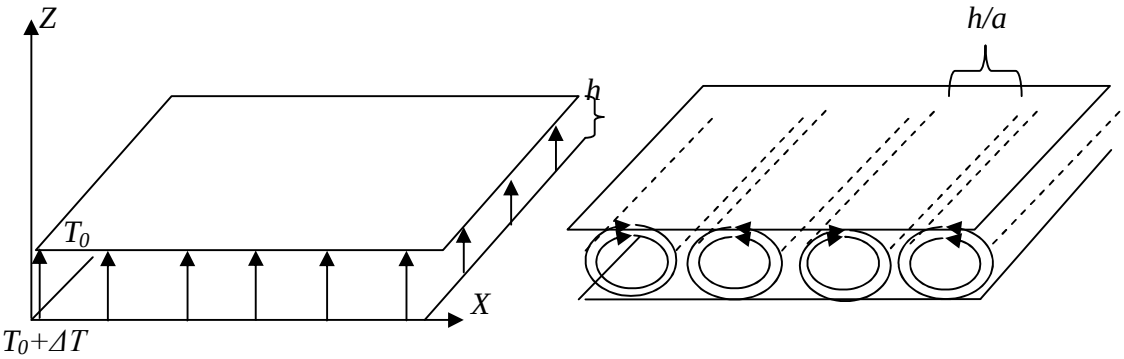


Figure 1.1 A heated fluid layer (Bernard experiment)

In a typical Bernard experiment, a fluid layer is heated from below in a gravitational field. The energy is transported through a fluid layer of depth *h*. The heated fluid at the bottom tries to rise, while the cold liquid at the top tries to fall. These motions are opposed by viscous forces. For small temperature differences ΔT, because of the viscosity no fluid motion takes place and the liquid remains at rest. Hence, heat is transported by uniform heat conduction. But at critical ΔT, convective or circulation flow occurs (Rayleigh-Bernard convection). The Navier-Stokes equations of motion of a viscous fluid are:

$$\frac{\partial \vec{V}}{\partial t} = \vec{F} - \nu \nabla^2 \vec{V} - \frac{1}{\rho} \nabla P + \left(\frac{\zeta}{\rho} + \frac{\nu}{3} \right) \vec{\nabla} \left(\nabla . \vec{V} \right), \tag{1.1}$$

where $\vec{V}$ is the velocity vector field, $\vec{F}$ is the body force per unit mass, P is the pressure, ρ is the density of the fluid, $\nu = \eta/\rho$ is the kinematic viscosity of the fluid, η is the dynamic viscosity, and ζ is the bulk viscosity which like dynamic viscosity is a positive quantity and depends upon the chemical nature of the compressed fluid. In equation 1.1, it is assumed that η and ζ are constant; otherwise the above equations of motion will have very complex form.

For an incompressible fluid, $\nabla . \vec{V} = 0$, and if the mass force field is potential as, for instance, the gravitational field, then we have $\nabla \times \vec{F} = 0$. The convective flow is assumed to be governed by the classic Navier-Stokes equations of motion of a viscous fluid:

$$\frac{\partial \vec{V}}{\partial t} + \vec{V} . \nabla \vec{V} = \vec{F} + \nu \nabla^2 \vec{V} - \frac{1}{D} \nabla P \ , \tag{1.2}$$

where for an incompressible fluid $\nabla . \vec{V} = 0$. Using equations (1.1) and suitable boundary conditions along with the continuity relation $\nabla . \vec{V} = 0$, the flow field $\vec{V}$ and P can be determined as functions of position in space and of time. Few mathematical solutions to this complicated set of nonlinear partial differential equations are known, except for simple geometries. For example, in two dimensions, the governing hydrodynamic equations are written in the non-dimensional stream function and purebred temperature form:

$$\begin{aligned} &\frac{\partial}{\partial t} \left(\nabla^2 \Psi \right) + \left(\frac{\partial \Psi}{\partial X} \right) \frac{\partial \nabla^2 \Psi}{\partial Z} - \left(\frac{\partial \Psi}{\partial Z} \right) \frac{\partial \nabla^2 \Psi}{\partial X} - \nu \nabla^4 \Psi - g\alpha \frac{\partial \Theta}{\partial X} = 0 \ , \\ &\frac{\partial \Theta}{\partial t} + \left(\frac{\partial \Psi}{\partial X} \right) \frac{\partial \Theta}{\partial Z} - \left(\frac{\partial \Psi}{\partial Z} \right) \frac{\partial \Theta}{\partial X} - \kappa \nabla^4 \Theta - \left(\frac{\Delta T}{h} \right) \frac{\partial \Psi}{\partial X} = 0 \ . \end{aligned} \tag{1.3}$$

Note that the stream function Ψ is defined such that the components of the flow velocity vector are given by: $V_x = -\frac{\partial \Psi}{\partial Z}, V_z = \frac{\partial \Psi}{\partial X}$. Θ is the departure of the temperature in the fluid from that which occurs when there is no convection present. This means $\Theta = T - T_0 - \Delta T\ (1 - z/h)$. The constant g is the gravity acceleration constant, $\alpha = -D^{-1}\ (\mathrm{d}D/\mathrm{d}T)$ is the coefficient of thermal expansion and κ represents the thermal conductivity. As the temperature difference ΔT increases the transport of energy from the lower to upper surface by heat conduction becomes unstable. At a critical value, the state of the fluid forms a pattern of stationary convection rolls (Figure 1.1). By the use of Fourier expansion the following solutions:

$$\Psi = \Psi_0 \sin\left(\frac{\pi a X}{h}\right)\sin\left(\frac{\pi Z}{h}\right),$$
$$\Theta = \Theta_0 \cos\left(\frac{\pi a X}{h}\right)\sin\left(\frac{\pi Z}{h}\right), \tag{1.4}$$

will be acceptable if they satisfy $R > R_c$, where,

$$R = \left(\frac{g\alpha h^3 \Delta T}{\nu\kappa}\right),$$
$$R_c = \left(\frac{\pi^4}{a^2}\right)\left(1 + a^2\right)^3.$$

R is called Rayleigh number and has the threshold minimum value of $R_c = 27\pi^4/4$ when $a^2 = 1/2$. From Figure 1.1 it is seen that parameter a is related to the horizontal size of the vortex. Beyond a greater temperature difference ΔT, the Rayleigh convection solution transition to chaotic motion is observed. The convection dynamics become time dependent and of seven spatial Fourier modes three modes persist. Lorenz studied these solutions which now depend on three functions of time, $x(t)$, $y(t)$, $z(t)$

$$\Psi(x,y,z,t)=\left[\frac{\sqrt{2}\left(1+a^2\right)}{a^2}\right]x(t)\sin\left(\frac{\pi aX}{h}\right)\sin\left(\frac{\pi Z}{h}\right),$$
$$\Theta(x,y,z,t)=\left[\frac{\sqrt{2}R_c}{\pi R}\right]y(t)\cos\left(\frac{\pi aX}{h}\right)\sin\left(\frac{\pi Z}{h}\right)-z(t)\left[\frac{R_c}{\pi R}\right]\sin\left(\frac{2\pi Z}{h}\right). \qquad (1.5)$$

These functions are different from spatial coordinates. Comparing these solutions to solutions (1.3) involves another vertical temperature variation. Lorenz ignored the spatial variations which are orthogonal to the ansatz (1.4) and obtained the three nonlinear differential equations of his famous model [15], describing the Bernard experiment:

$$\begin{aligned}\frac{dx}{dt'}&=\sigma(y-x),\\ \frac{dy}{dt'}&=x(\rho-z)-y,\\ \frac{dz}{dt'}&=xy-\beta y.\end{aligned} \qquad (1.6)$$

In 1963 Lorenz observed that these three coupled first-order nonlinear differential equations could lead to completely chaotic trajectories. These equations can be written in dimensionless time $t'=\left(\frac{\pi^2}{h^2}\right)\left(1+a^2\right)\kappa t$. The parameter σ is a non-dimensional ratio of viscosity to thermal conductivity (ν/κ), $\rho = R/R_c$ is a non-dimensional temperature gradient which is related to Rayleigh number, and $\beta = 4\left(1+a^2\right)^{-1}$ is a geometric factor. Lorenz used $\sigma = 10$ and $a^2 = 1/2$, corresponding to minimum critical Rayleigh number. Thus, in this case $\beta = 8/3$. Each differential equation describes the rate of change for a variable $x(t)$ proportional to the circulatory fluid flow velocity, a variable $y(t)$ characterizing the temperature difference between ascending and descending fluid elements, and a variable $z(t)$ proportional to the deviation of the vertical temperature profile from its equilibrium value. The system Lorenz used to model the dynamics of weather differs from Hamiltonian systems mainly by its dissipativity. A dissipative system is

not conservative, with an external control parameter that can be tuned to critical values causing the transitions to chaos. In other word, from these equations, it can be derived that an arbitrary volume element of some surface in the corresponding phase space contracts exponentially in time. This can be visualized by numerical calculations of the trajectories generated by the three equations of the Lorenz model. Under certain conditions, a particular region in the three-dimensional phase space is attracted by the trajectories, making one loop to the right, then a few loops to the left, then to the right again. The paths of these trajectories depend very sensitively on the initial conditions. Tiny deviations of their values may lead to paths which soon deviate from the old one with different numbers of loops. Because of its strange image, which looks like the two eyes of an owl, the attracting region of the Lorenz phase was called a strange attractor. Chaotic regimes for the Lorenz system are studied and analyzed in Chapter 5. As a final point about the Navier-stokes equations of motion of a viscous fluid, notice that the viscous term on the left-hand side of equation (1.1) is linear and is based on the assumption of a Newtonian fluid. One can imagine that if one goes beyond the study of the Navier-Stokes equation to include nonlinear viscous fluids (non-Newtonian fluids), or elastoplastic materials, there is a vast array of nonlinear and chaotic phenomena can be discovered. In all these system equations nonlinearity is a necessary, but not sufficient, condition of chaos. It is a necessary condition, because linear differential equations can be solved by well-known mathematical procedures (Fourier transformations) and do not lead to chaos.

As an example of a dissipative chemical reaction system in which chaotic motion has been studied experimentally is the Belousov-Zhabotinsky reaction [16]. In this chemical process an organic molecule is oxidized by bromate ions, the oxidation being catalyzed by a redox system. The rates of change for the concentrations of the reactants in a system of chemical

reactions are again described by a system of nonlinear differential equations with a nonlinear function. The variable which signals chaotic behavior in the Belousov-Zhabotinsky reaction is the concentration of ions in the redox system. Experimentally, irregular oscillations of these concentrations are observed with a suitable combination of the reactants. The oscillations are indicated by separated colored rings. This separation is a fine visualization of nonlinearity evolutions which satisfy the superposition principle. In this case the oscillating rings penetrate each other in superposition. The corresponding differential equations are autonomous, that is, they do not depend on time explicitly.

To visualize the numerical solutions it is often convenient to study the flow in a dynamical system described by differential equations of motion via discrete equations which construct the intersecting points of the trajectories with the (*N-1*)-dimensional Poincare map in the corresponding *N*-dimensional phase space. The constructed points are denoted by $X_1, X_2, \ldots, X_n, X_{n+1}, \ldots$ with increasing time points *n*. The corresponding equation has the form $X_{n+1} = F(X_n, c)$ for the successor points. The classification of conservative and dissipative systems can be generalized from flows to Poincare maps. A discrete map equation is called dissipative if it leads to a contraction of volume in phase space. A famous example of a discrete map is a logistic map with many applications in the natural sciences as well as the social sciences. The basic concepts of complex dynamical systems from nonlinearity to chaos can be illustrated by a logistic map. Mathematically, a logistic map is defined by a quadratic recursive map: $X_{n+1} = \mu X_n(1 - X_n)$ of the interval $0 \le X_n \le 1$ onto itself with control parameter μ varying between $0 \le \mu \le 4$. The function values of the sequence $X_1, X_2, \ldots, X_n, X_{n+1}, \ldots$ can be calculated easily. For $\mu \le 3$ the sequence converges towards a fixed point. If μ is increased beyond a critical value μ_k, then the values of the sequence jump periodically between two values after a

certain time of transition. If μ is increased further beyond a critical value μ_2, the period length doubles. If μ is increased further and further, then the period doubles each time with a sequence of critical values of μ. Beyond the bifurcating critical values the development becomes more and more irregular and chaotic.

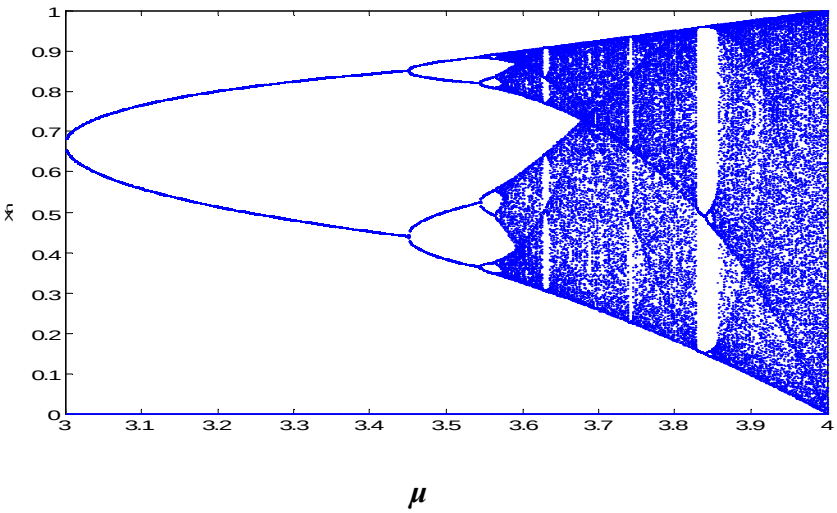


Figure 1.2 Bifurcation diagram, X_n vs. μ for logistic map $X_{n+1} = \mu X_n(1 - X_n)$.

The sequence of period doubling bifurcations which is illustrated in Figure 1.2 is governed by a law of constancy recognized by Feigenbaum as a universal property for a whole class of functions (the Feigenbaum-constant). It is rather astonishing that a simple mathematical law like the logistic map produces a complexity of bifurcations and chaos for possible developments. A necessary, but not sufficient, reason is the nonlinearity of the equation. In this context, the degrees of increasing complexity are defined by the increasing bifurcations which lead to chaos as the most complex and fractal scenario. Each bifurcation illustrates a possible branch of solution for the nonlinear equation. Physically, they denote phase transitions from a state of equilibrium to new possible states of equilibrium. If equilibrium is understood as a state of symmetry, then a phase transition means symmetry breaking which is caused by fluctuational forces.

CHAPTER 2

MEASURES OF ORDER IN DYNAMIC SYSTEMS

2.1 Introduction

The usefulness of the Lyapunov exponent as a metric to characterize the dynamic patterns of discrete and continuous systems with chaotic, periodic, and quasi-periodic responses is studied in a few dynamic systems. We have used Lyapunov exponent to evaluate the order and disorder in nonlinear dynamic systems behavior.

In nonlinear chaotic systems small changes in initial conditions will lead eventually to large changes in the behavior of the system. Nevertheless the system remains stable due to its deterministic nature. This occurs because the chaotic attractors with fractal geometry are confined in a certain region in phase space. There are several metrics to measure chaos, depending on what one wants to characterize in the chaotic trajectory. Certain quantities such as Lyapunov exponents, fractal dimensions, K-entropy, algorithmic complexity and power spectrum analysis have been used as screening tools to detect chaos in nonlinear systems. But methods like Fourier transform, and the resulting power spectrum density, fails to distinguish between chaos and noise, because both phenomena are broadband. Two practical approaches to evaluate the order and disorder in the output signal of nonlinear systems output signal are the algorithmic complexity measure [17, 18, and 19] and the Lyapunov exponent spectrum [20, 21, and 22]. Mapping the system output signal to a binary string and calculating the complexity measure of the time-series data, characterizes strange attractors by the pattern formation in phase space attractors. In this chapter we review the Lyapunov exponent concept and its analytical

methods using Gram-Schmidt Reorthonormalization (GSR) and No Rescaling No Orthogonalization (NRNO). In chapter 3 and 4 we apply these methods to a single Josephson junction and Lorenz system. After introducing complexity concept in chapter 4, the rest of the chapter deals with the comparing the fundamental concept of measuring chaos in dynamical systems through the Lyapunov exponents and algorithmic complexity measure.

2.2 Lyapunov Characteristic Exponents

2.2.1 Introduction

Lyapunov exponents provide the single most important quantitative characterization of the exponential divergence of initially nearby trajectories which is the hallmark of chaos in complex systems. The concept of invariant exponents in the study of the stability of nonlinear differential equations was first introduced in 1889 by a Russian mathematician Sonya Kovalevskaya. This theory later developed fully in 1892 by another Russian mathematician, Alexander Mikhailovich Lyapunov. Recent applications of these exponents include the connection between chaotic dynamics and transport theory in statistical mechanics.

2.2.2 Lyapunov Exponent Definition

A necessary condition for the dynamics in the region of the solution, $\vec{X}(t)$ to be chaotic is that $\lambda > 0$. This condition, however, is not sufficient. The quantity λ defined by $\lambda = \frac{1}{n}\log_2\left(\frac{\left\|\delta\vec{X}(t)\right\|}{\left\|\delta\vec{X}(0)\right\|}\right)$, is called a Lyapunov characteristic exponent. To show that $\lambda > 0$ is a necessary condition and not a sufficient one considers the solution of $\frac{dX(t)}{dt} = aX(t) + b$ satisfying $X(0) = c$. The Lyapunov exponent $\lambda = a$, for all a both when $a = 0$ and $a \neq 0$. The solution is clearly not chaotic, even if $\lambda > 0$. For chaos the additional property of the dynamics

which generally insures chaos is that there must be a contracting direction producing bounded motion. Lyapunov exponents are a generalization of the usual characteristic exponents which are obtained to assess the stability of fixed points. In that case the special solution $\vec{X}(t)$ is simply a constant, $\vec{X}_0$ and the solutions of the variational equations are easily obtained. It should be emphasized that, in general, the Lyapunov exponent depends on the particular solution of the nonlinear equation which is used in the variational equation. Put another way, the quality or existence of the chaos may depend on the region of the phase space through which $\vec{X}(t)$ passes. In a special case, if the dynamics is ergodic, then the exponent is independent of $\vec{X}(t)$, since it approaches arbitrarily close to all points in the relevant phase space. But, ergodic systems are not as frequently found as theoreticians used to believe, and the dynamics in phase space often has much more structure than previously expected.

2.2.3 Lyapunov Exponent (LE) Estimation

In the following sections we review the fundamentals of two dominating methods in the computation of Lyapunov characteristic exponents, that is, the GSR and NRNO methods. But first we consider the principle behind evaluating the divergence of trajectories of dynamical systems (as a function of time) by a single Lyapunov exponent, and later will consider a more practical approach by rescaling and renormalization of the trajectories. Finally, we will study strange attractors in *N*-dimensional phase space, as they originate from analytical models or are reconstructed from experimental time series data.

Consider any two nearby divergent trajectories, originating from a three-dimensional flow (i.e. trajectories in continuous time as described by differential equations), as illustrated in Fig. 2.1. Based on its definition, the Lyapunov exponent (LE) estimate is easy to understand and

compute, but it is not very accurate because of the exponential growth of the distance $\delta\vec{X}(t)$. Instead, we can compute an average exponent by tracking not one but many nearby trajectories. Once the difference between the trajectories is too large to handle $\delta\vec{X}(t)$, another nearby trajectory is selected and tracked. This is repeated until the end of data in the time series is reached. For *N* segments of the nearby trajectories, the average is given whereby the superscript enclosed in parentheses signifies the successive trajectory selections taken in the computation. By averaging over *N* trajectory segments, we reduce the error in the LE estimate. For a discrete time map the growth of the difference between the two nearby trajectories over a time period can be described by $\delta\vec{X}_n = \delta\vec{X}_0 e^{n\lambda}$ and the exponent can be computed from $\lambda = \frac{1}{n}\ln\left(\frac{\|\delta\vec{X}_n\|}{\|\delta\vec{X}_0\|}\right)$.

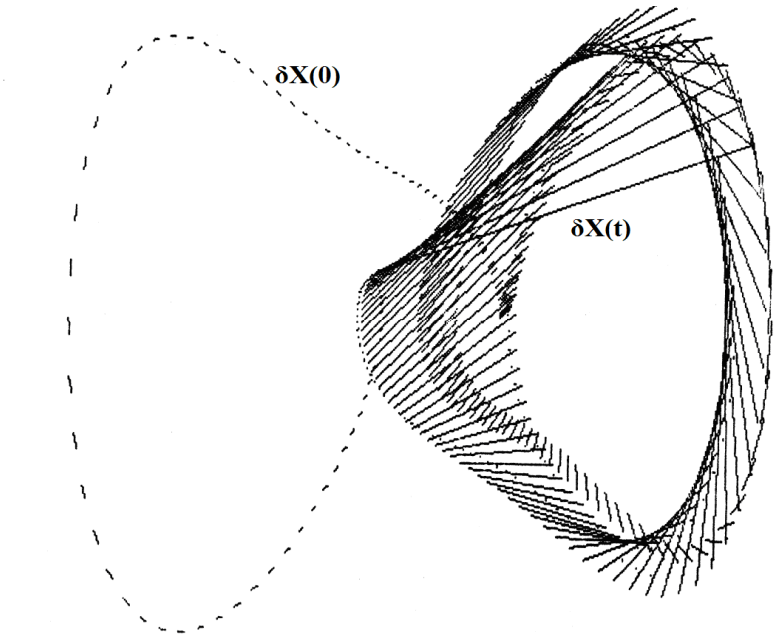


Figure 2.1 Exponential divergence of two nearby trajectories for a dynamical flow $\delta\vec{X}(t) = \delta\vec{X}(0)e^{\lambda^+ t}$, with future prediction time $t_p = \frac{1}{\lambda^+}\ln\left(\frac{\|\delta\vec{X}(t)\|}{\|\delta\vec{X}(0)\|}\right)$

The importance of the above approach to compute the LE is that the stability of the dynamical system can be determined without actually knowing and solving the underlying

equations explicitly. This occurs when we obtain a chaotic time series from a dynamical system, reconstruct its strange attractor in the corresponding pseudo-phase space, and then compute the Lyapunov exponents from the reconstructed strange attractor directly, without its explicit mathematical model. Another important reason for using the Lyapunov exponent as a characteristic measure of a dynamical system is its invariance to rescaling, shifts, and other transformations of data such as the imprecise reconstruction of a strange attractor from a time series. Thus, any smooth invertible reparameterization of phase space can, in the worst case, change the ratios by finite factors.

It is important to realize that the averaging of LE can be interpreted as the average divergence rate if and only if the individual LE is invariant to the phase-space location. In this case, one can average over long time periods, thus producing the global Lyapunov exponent [23]. In practice, the partial LE's are not invariant due to either nonstationarity of the system or multifractality of the process, and the averaging can only be done over short periods. This defines local Lyapunov exponents. Consequently, one can propose that this lack of invariance of Lyapunov exponents be used as a new test for multifractality of a strange attractor, similar to the spread of the Rényi fractal dimension spectrum and the Mandelbrot fractal dimension spectrum.

2.2.4 Lyapunov Exponents for 1-Dim map

The global Lyapunov exponent of a discrete one dimensional system can be derived if the map is known explicitly. On the other hand, if the map is given implicitly through a time series, we must compute the exponents numerically. The concept can be better understood if we first study a 1-Dim map and its corresponding single LE, as a preamble to N-dimensional maps with N Lyapunov exponents, representing the stretching and/or shrinking either along each dimension or each principal component in the phase space.

Consider an iterative 1-Dim map consisting of scalars $X_{n+1} = F(X_n)$ where the map $F(X_n)$ is used to predict the next value X_{n+1} from the current value X_n, thus producing a discrete time series $\{X_n\}$, generated at regular intervals labeled *n*. The system is deterministic, as X_{n+1} is always the same if X_n is the same. A Lyapunov exponent of a 1-Dim map $X_{n+1} = F(X_n)$ for an initial condition X_0, measures the average error growth rate per iteration or equivalently the average loss of information during successive iterates near X_0. If the map is known explicitly, it is possible to calculate the Lyapunov exponents explicitly. Let us consider two nearby points X_0 and $X_0 + \delta X_0$. The n^{th} iterate relates the n^{th} value in the series $\{X_n\}$ to the initial value X_0, i.e. $X_n = F(X_{n-1}) = F(F(...F(X_0)...))) = F^n(X_0)$. Since $F(X_n) : [0,1] \to [0,1]$, the local Lyapunov exponents with respect to X_0 can be written as,

$$\exp(\lambda X_0) = \frac{\delta F}{\delta X} = \frac{F(X_0 + \Delta) - F(X_0)}{\Delta},$$

$\frac{\delta F}{\delta X}$ denotes the first derivative of the n^{th} iterate of the map $F(X_n)$. After the n^{th} iteration the Lyapunov exponent with respect to X_0 is:

$$\exp(n\lambda X_0) = \frac{\delta F}{\delta X} = \frac{F^n(X_0 + \Delta) - F^n(X_0)}{\Delta}.$$

For the exponential error growth between the points, we also have, taking the e-base logarithm:

$$\lambda(X_0) = \lim_{n\to\infty}\left(\frac{1}{n}\log_2\left|\frac{dF^n(X_0)}{dX}\right|\right).$$

By chain differentiation, $F^n(X_0) = F(F(...F(X_0)...)))$ we have:

$$\frac{dF^n(X_0)}{dX} = \frac{dF(X)}{dX}\Big|X_{n-1}\,\frac{dF(X)}{dX}\Big|X_{n-2}\cdots\frac{dF(X)}{dX}\Big|X_0$$

$$\lambda(X_0) = \lim_{n\to\infty}\left(\frac{1}{n}\log_2\left|\frac{dF^n(X_0)}{dX}\right|\right) =$$

$$\lim_{n\to\infty}\left(\frac{1}{n}\log_2\left|\frac{dF(X)}{dX}\Big|X_{n-1}\,\frac{dF(X)}{dX}\Big|X_{n-2}\cdots\frac{dF(X)}{dX}\Big|X_0\right|\right) =$$

$$\lim_{n\to\infty}\left(\frac{1}{n}\sum_{j=0}^{n-1}\log_2\left|\frac{dF(X)}{dX}\Big|X_j\right|\right).$$

Since the first derivative of a one-dim map is constant, the global Lyapunov exponent can be computed by dropping both the limiting condition and the sum. In fact the global Lyapunov exponent is equal to the local Lyapunov exponent in this case. However, for non-constant first derivatives, the flow may exhibit a transient in the trajectories, lasting from the initial condition to the iteration *m*, where the trajectories will become ergodic. In order to avoid incorrect values for the divergence of the trajectories, we must ignore the transient solutions from first iterates up to *m*. The absolute value assures that the logarithm is real and that its sign is inconsequential to the growth. That is, when the ratio is negative, the two nearby points can be considered interchanged in order only, but not in their growth. Since λ is proportional to the logarithmic measure of the rate, if $\lambda > 0$, the map has sensitive dependence of the initial conditions, and if $\lambda = 0$, the map is insensitive to the initial conditions. Once again, since the rates may vary from point to point, we should average the local divergence over a large number of points to obtain the global Lyapunov exponent. The base of the logarithm can be either $e = 2.71$, which gives the λ in terms of the loss of information in nats/iteration, or base 2 with bits/iteration. The latter interpretation is important because it signifies the spread of Shannon uncertainty. If we take $\lambda = 1$ bit/iteration and the accuracy of the initial value to be $\varepsilon_0 = 0.01$ (or 1%), then the accuracy

would grow to $\varepsilon_1 = \varepsilon_0 2^{\lambda} = 0.01 \times 2 = 0.02$ after one iteration and to $\varepsilon_2 = 0.02 \times 2 = 0.04$ after two iterations, and it exceeds the entire range $\varepsilon_4 = 0.01 \times 2^4 = 0.16$ after only four iterations. This means that, if X_n is coded as a binary number, each iteration would render the least significant bit uncertain, or one bit per iteration. This idea of the loss of information per iteration is the source of unpredictability. For numerical solutions we choose a logistic map with n as the maximum number of iterates and m as the number of discarded initial transient iterates, with $m < n$. Figure 2.2 (a) illustrates the case when transient responses are included in bifurcation map and assuming that the parameters are: m=0, n=1000 and X_0=0.3. This calculation process is repeated for $0 \leq \mu \leq 4$ with increments of $\Delta\mu$=0.001. Bifurcation map without 500 transient solutions at each step is shown in 2.2 (b). In addition, in the same range of control parameter, we found the values of the Lyapunov exponent. This is shown in Figure 2.2 (c). The negative LE values correspond to stable states and the positive values are for chaotic states.

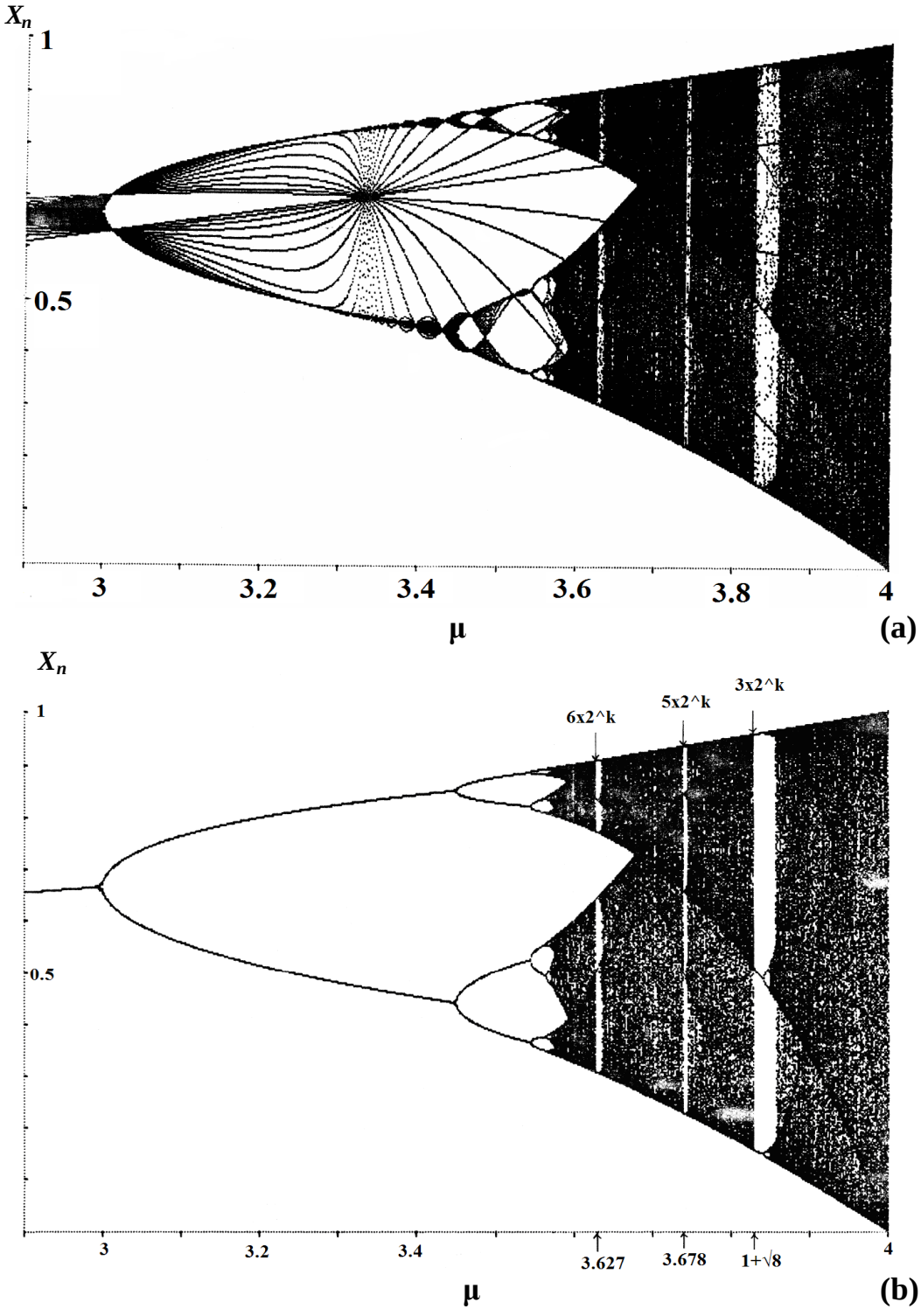


Figure 2.2 (a) Logistic bifurcation map including transient solutions (b) Logistic bifurcation map without transients solutions (c) Lyapunov exponent λ vs. μ.

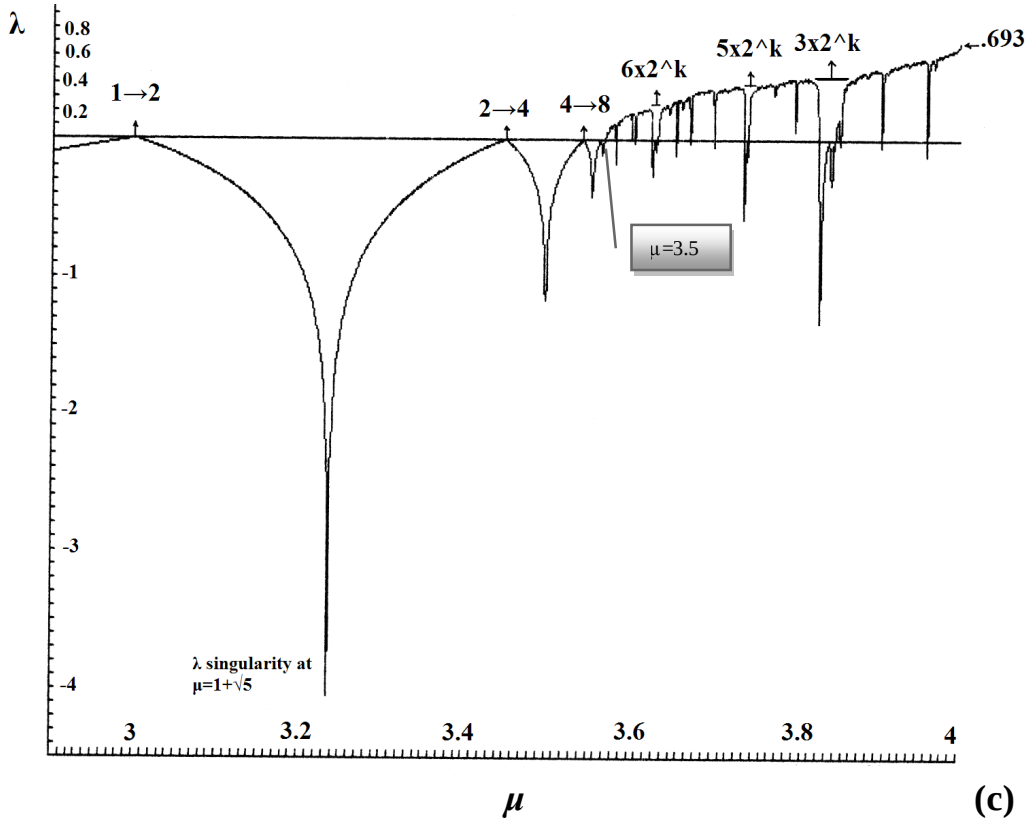


Figure 2.2, cont. (a) Logistic bifurcation map including transient solutions (b) Logistic bifurcation map without transients solutions (c) Lyapunov exponent λ vs. μ.

Comparing Figure 2.2 (b) and (c) it is seen that LE is negative for all the stable points and for the reemerging order within chaos (e.g., the period-3 band). It reaches 0 at the bifurcation points. Figure 2.2 (b) also shows how the behavior of λ just prior to the bifurcation points could be used to predict the point where λ attains its maximum value at which total chaos is reached ($\mu = 4$) and beyond which instability commences. It is also possible to compute an analytical expression for the singular points of the Lyapunov function. For example, for two-cycle behavior we will find the range of control parameter, μ. Let a_1 and a_2 be the solution to the two-cycle periodic solution to Logistic map. We can write:

$$a_1 = \mu a_2 (1 - a_2),$$
$$a_2 = \mu a_1 (1 - a_1),$$

$$\frac{dF^P}{dX}\Big|_{x=a_i} = \frac{dF}{dX}\Big|_{x=a_1}.\frac{dF}{dX}\Big|_{x=a_2}....\frac{dF}{dX}\Big|_{x=a_p} = \prod_{j=1}^{P}\frac{dF^P}{dX}\Big|_{x=a_j},$$

$$\left|\frac{dF^P}{dX}\Big|_{x=a_i}\right| = \prod_{j=1}^{P}\left|\frac{dF^P}{dX}\Big|_{x=a_i}\right| .$$

Let $F(X) = \mu X(1-X)$, then $\frac{dF}{dX} = \mu(1-2X)$.

a_1, a_2 are the solutions of $F(F(X)) = 0$, that is:

$$\mu[\mu X(1-X)][1-\mu X(1-X)] = 0.$$

or,

$$a_1 = \frac{1}{2\mu}[(1+\mu) + \sqrt{\mu^2 - 2\mu - 3}],$$

$$a_2 = \frac{1}{2\mu}[(1+\mu) - \sqrt{\mu^2 - 2\mu - 3}].$$

The condition for stability is,

$$\left|\frac{dF^P}{dX}\Big|_{x=a_j}\right| < 1,$$

and

$$\frac{dF}{dX}\Big|_{x=a_p} = \mu(1-2a_p),\ p = 1,\ 2,$$

$$\prod_{j=1}^{P}\left|\frac{dF^P}{dX}\Big|_{x=a_j}\right| < 1\ ,\ j = 1,\ 2,$$

$$-1 < \mu^2(1-2a_1)(1-2a_2) < 1,$$

with solutions $\mu > 4, \mu < -1$ and $1-\sqrt{6} < \mu < 1+\sqrt{6}$. The acceptable range for stability of 2-cycle solutions is $3 < \mu < 1+\sqrt{6}$. Now we can compute the value of the Lyapunov exponent for a 2-cycle oscillation, starting from initial point X_0 as follows

$$\lambda(X_0)=\lim_{n\to\infty}\left(\frac{1}{n}\ln\left|\frac{dF^n(X_0)}{dX}\right|\right)$$

$$=\lim_{n\to\infty}\left(\frac{1}{n}\ln\left|\frac{dF(X)}{dX}\Big|_{X_0}\cdot\frac{dF(X)}{dX}\Big|_{X_1}\cdots\frac{dF(X)}{dX}\Big|_{a_1}\cdot\frac{dF(X)}{dX}\Big|_{a_2}\right|\right)=$$

$$\lim_{n\to\infty}\frac{1}{n}\left(\ln\left|\frac{dF(X)}{dX}\Big|_{X_0}\cdot\frac{dF(X)}{dX}\Big|_{X_1}\cdots\frac{dF(X)}{dX}\Big|_{X_m}\right|+\ln\left|\frac{dF(X)}{dX}\Big|_{a_1}\cdot\frac{dF(X)}{dX}\Big|_{a_2}\cdots\frac{dF(X)}{dX}\Big|_{a_1}\frac{dF(X)}{dX}\Big|_{a_2}\right|\right)$$

$$=\lim_{n\to\infty}\frac{1}{n}\left((\frac{n-m-1}{2})\ln\left|\frac{dF(X)}{dX}\Big|_{a_1}\frac{dF(X)}{dX}\Big|_{a_2}\right|\right)=\frac{1}{2}\ln\left|\frac{dF(X)}{dX}\Big|_{a_1}\frac{dF(X)}{dX}\Big|_{a_2}\right|$$

$$=\frac{1}{2}\ln\left|(-1-\sqrt{\mu^2-2\mu-3})(-1+\sqrt{\mu^2-2\mu-3})\right|=\frac{1}{2}\ln\left|(-\mu^2-2\mu-3)\right|,$$

where $3<\mu<1+\sqrt{6}$. This expression has a singularity at $\mu=1+\sqrt{5}$. This analytical solution is verified by numerical solution depicted in the Lyapunov exponent plot. Through the same process, theoretically one can calculate the analytical form of Lyapunov exponent at the regions of $\mu_k=2^k n$ periodic behavior. It is seen that the Lyapunov exponent λ summarizes the dynamics of the logistic map as a function of μ. For negative λ, there is no sensitive dependence on initial conditions and the stability of the system is unconditional, leading to the limit points and limit cycles. This continues from $\mu=0$ to $\mu=3.57$, where chaos starts. This is the transition point between chaos and multi-periodicity, and is called accumulation point. At all the bifurcation points μ_k the Lyapunov exponent becomes zero, $\lambda = 0$. The exponent also becomes negative within all the reemergence of order in the chaotic region (within the period-3 region). In the chaotic region, λ increases to ln(2) at $\mu=4$. This can be shown analytically as follows:

$$\langle\lambda\rangle=\int\left(\ln\left|\frac{dF(X)}{dX}\right|\right)P(X)dX.$$

For logistic map $F(X)=\mu X(1-X)$, and $\frac{dF}{dX}=\mu(1-2X)$, with probability distribution function normalized in [0, 1]: $P(X)=\frac{1}{\pi\sqrt{X(1-X)}}$.

The logistic map global LE for $\mu = 4$ is computed as follows:

$$\langle\lambda\rangle = \int\left(\ln\left|\frac{dF(X)}{dX}\right|\right)P(X)dX = \int_0^1 \frac{\ln|4(1-2X)|}{\pi\sqrt{X(1-X)}}dX =$$
$$= \int_0^1 \frac{\ln|4|}{\pi\sqrt{X(1-X)}}dX + \int_0^1 \frac{\ln|1-2X|}{\pi\sqrt{X(1-X)}}dX .$$

But since $P(X) = \frac{1}{\pi\sqrt{X(1-X)]}}$ is normalized in [0, 1], that is

$$\int_0^1 P(X)dX = \int_0^1 \frac{dX}{\pi\sqrt{X(1-X)}} = 1 ,$$

Then the first integral is ln(4). In the second integral we substitute $(1-2X) = \sin\theta$,

$$\frac{dX}{\sqrt{X(1-X)}} = d\left(-\sin^{-1}(1-2X)\right) = -d\theta .$$

Then

$$\int_0^1 \frac{\ln|1-2X|}{\pi\sqrt{X(1-X)}}dX = \frac{2}{\pi}\int_{\frac{\pi}{2}}^{0} -d\theta\ln(\sin\theta) = (\frac{2}{\pi})(\frac{\pi}{2})\ln 2 = \ln 2 .$$

Therefore,

$$\langle\lambda\rangle\big|_{\mu=4} = \ln 4 - \ln 2 = \ln 2 .$$

This is confirmed numerically at the last point in Figure 2.2 (c). Since the logistic map is a parametric quadratic map, there is a single global Lyapunov exponent for a given initial condition X_0 and μ, $\lambda(X_0, \mu)$. However, the logistic map exhibits a transient in the trajectories from the initial condition to the point where the trajectories become ergodic. For most of the continuous flows, the convergence of λ is established after a number much larger iterations, ignoring the transient states.

2.3 Computing Lyapunov Exponents

When the dynamics are governed by a set of *N* ordinary differential equations $\frac{d\vec{X}}{dt} = \vec{F}(\vec{X}, \vec{X}_0, \vec{C})$, the spectrum of Lyapunov exponents is related to the integration of the linearly perturbed dynamics, $\vec{\eta}(t)$, about a reference solution $\vec{X}^*(t)$, as in $\frac{d\vec{\eta}}{dt} = A(\vec{X}^*(t)).\vec{\eta}$. The matrix of partial derivatives $A = \nabla\vec{F}(\vec{X}^*)$ is time-dependent, because it depends in general on the reference solution $\vec{X}^*(t)$. The solution at time $t = \tau$ is $\vec{\eta}(\tau)$ and can then be written formally in the form $\vec{\eta}(\tau) = \Phi(\vec{\eta}*(t)).\vec{\eta}(0)$, where Φ is an *N* by *N* matrix, $\vec{\eta}^*(t)$ is the linearly perturbed dynamics solution about reference trajectory $\vec{X}^*(t)$ and $\vec{\eta}(0)$ is the initial perturbation. A formal definition of the Lyapunov exponent λ_i follows from the construction of a positive and symmetric matrix $\tilde{\Phi}\Phi^{\frac{1}{2\tau}}$ whose eigenvalues are μ_i. Then the Lyapunov exponents are defined by: $\lambda_i = \lim_{\tau\to\infty}(\log\mu_i)$. This formal definition does not provide an obvious and practical way to determine the complete set of λ_i from numerical or experimental data.

There are several implementations of algorithms for calculating Lyapunov exponent spectrum including those by Wolf, Shimada–Nagashima, Rosenstein–Kantz, and an eigenvalue-technique based on estimating the local Jacobian matrices and a group theoretical-based NRNO (No-Rescaling-No-Orthogonalization) approach [21]. Some of the recently invented algorithms have a built-in check for chaos and noise. Historically, the first practical algorithm was provided by Wolf et al. [20]. Their implementation was designated primarily for the principal Lyapunov exponent λ_1 and also for the sum of the first two Lyapunov exponents $\lambda_1 + \lambda_2$ extracted from a time series. QR factorization and Singular Value Decomposition (SVD) methods [24] require

frequent renormalization to overcome the exponential growth of the separation vector between the fiducial and nearby trajectories. Also they use reorthogonalization to overcome the exponential collapse of initially orthogonal separation vectors onto the direction of maximal growth. The existing continuous versions of the QR and SVD methods suffer from the additional disadvantage of being unable to compute the partial Lyapunov spectrum using a fewer number of equations/operations than required for the computation of the full spectrum. Further, the continuous version of the SVD method breaks down when computing degenerate Lyapunov spectra. The widespread perception that some form of explicit rescaling and reorthogonalization is necessary lies at the heart of most methods for computing Lyapunov exponents. Although very popular, this algorithm is not very robust for real time applications involving real-time data series. Thus, it cannot be recommended for practical calculations in noisy environments.

Typically, one has a set of discrete time sampled data $X_k = X(t = ks)$, where *X(t)* is a measured state variable and *s* is the sampling time. One numerical technique for a set of data $\{\cdots X_{k-1}, X_k, X_{k+1}, \cdots\}$ is to construct a set of vectors in an embedding space of *N* dimensions, $(X_i, X_{i+1}, \cdots, X_{j+N-1}, \cdots)$. *N* is chosen at least as large as the dimension of the space of the chaotic attractor. Two methods are then used to calculate Lyapunov exponents. One is based on calculating the change in a small hyper-volume as discussed above as the dynamics evolves. For example, an $M \le N$-dimensional volume will change on average according to $V(t)=V_0\exp\{(\lambda_1+\lambda_1 \ldots+\lambda_M)t\}$. This method has been adopted by Wolf et al. Also, we have to emphasize that computing Lyapunov exponents requires many special considerations. For example, the size of the volume element in which the divergence of the adjacent trajectories is evaluated must be selected properly with respect to the density of trajectories in the strange attractor. Also, the estimates of Lyapunov exponents may not be reliable if the number of data points is too small.

Furthermore, noise has a destructive impact on Lyapunov exponent calculations. In any implementation of an algorithm, there are many other problems to be solved, including the selection of a reference trajectory, following the neighboring trajectories, reorthogonalization and reorthonormalization frequency, efficient storage, and user interface. In addition, the major evolution of the algorithms was prompted by the need to improve their robustness to noise.

Estimating Lyapunov spectra from a chaotic time series was provided by Frøyland and Alfsen [25], Frøyland [26], Sano and Sawada [27], Eckman et al. [28], Darbyshire and Broomhead [29], Nagashima, Baba and Sprott [30], Kantz and Schreiber [31]. Their implementations produce very good results if the time series is a good representation of the system dynamics, with a minimal contamination by noise. Estimation of Lyapunov exponents from noisy strange attractors was the next step in the evolution of this algorithm, with the inclusion of the two most common techniques, the Gram-Schmidt reorthonormalization and the QR decomposition. The developments included contributors Bryant et al. [32], Parlitz [33], and Nychka et al. [34]. Yao and Tong [35] also used the Kullback-Leibler measure as a distance over conditional distributions given past distributions. The study of various algorithms and their improvements with respect to noise immunity and the reduction of spurious exponents still continue today. In most of these methods, several factors can raise questions as to the accuracy of the exponents. These factors include: (1) ill-conditioned matrices, (2) a small number of data points, $N < 10^4$ (3) the accuracy of data, (4) the peculiar geometry of the attractor, and (5) spurious exponents when N is too high. Reliable algorithms for data experimental calculation of all λ_i in the Lyapunov spectrum are still under study. The measured Lyapunov exponents are not very precise and one should use them with some suspicion, especially where the dimension of the attractor is six or higher. Sections 2-4 through 2-7 provide the theoretical background of a

modified version of the Wolf et al. approach, referred to here as the Gram-Schmidt Reorthonormalization (GSR) method, followed by the group theoretic estimation of Lyapunov exponents (the NRNO) method.

2.4 GSR general Formalism

Lyapunov characteristic exponents give a complete indication of the nature of the flow in the neighborhood of a solution $\vec{X}(t)$. In *N*-dimensional systems, the stretching and contraction along the principal axis in phase space produce a spectrum of Lyapunov exponents. The Lyapunov exponents λ_i *(i = 1, 2, 3,…N)* measure the exponential growth of the principal axes of this generalized ellipsoid. The exponent λ_1 measures the largest of these exponential growths, because most initial vectors ($\vec{Z}_0$) will tend to orient along this direction as time increases. Therefore we cannot determine the exponent associated with another axis of the ellipsoid simply by considering another initial vector, since it will almost certainly become captured by the direction of most rapid growth. For example, the deformation of a sphere along a reference trajectory, i.e., the stretching and contraction of the principal directions, as well as the rotation of the principal directions, for a 3-dimensional flow is illustrated in Figure 2.3. This shows an initially spherical region around $\vec{X}_0$ becoming distorted into some ellipsoidal shape.

The general assumption is that the initially orthogonal directions remain orthogonal after a time interval τ. In practice, this assumption may not be satisfied, and the Gram-Schmidt reorthogonalization and renormalization of the directions must be done after each time interval τ. Because of the stretching along the first principal direction $\vec{Z}^{(1)}$, the corresponding Lyapunov exponent, λ_1 is positive in bits per iteration if the logarithm is calculated in base two units. If any exponent is positive, the system cannot be stable in the sense of the limit point or cycle stability.

However, the system may still be stable in the chaotic sense, if the flow has a folding capability, hence the strange attractor appears.

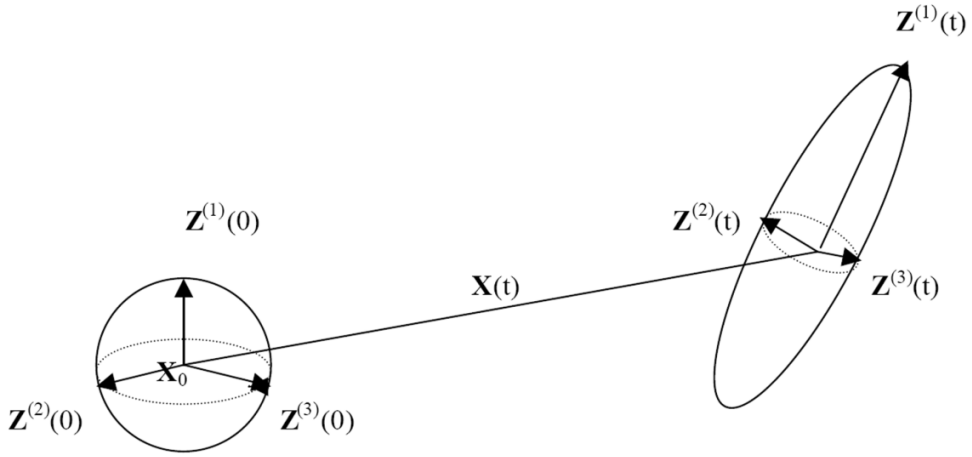


Figure 2.3 Divergence of trajectories from a small sphere of initial conditions

Next, we describe the general formalism of the Gram-Schmidt reorthonormalization method. This is done by solving dynamical systems equations along with variational equations to determine the Lyapunov exponents in an *N*-dim flow. Because we can rarely obtain the solution of the variational equations, the Lyapunov exponent must usually be obtained numerically. One of the numerical problems which are encountered is that any exponential solution of the variational equation soon becomes too large to treat computationally. This difficulty can be overcome by making use of the linearity of the variational equation. Let $\vec{X}(t) = \overline{\overline{T}}^{t}_{\vec{X}_0}$ be the solution of

$$\frac{d\vec{X}}{dt} = \vec{F}(\vec{X}, \vec{X}_0, \vec{C}), \quad \vec{X} \in \Re^N,$$

with the initial condition $\vec{X}_0$, and $\vec{Z}^{(i)}$ represent the solution of the variational equation as

$$\frac{d\vec{Z}^{(i)}(t)}{dt} = \overline{\overline{J}}.\vec{Z}^{(i)}(t).$$

Since the structure of the coupled differential equations is of a special form, they may also turn out to be useful for analytic studies of the evolution in tangent space. Using the tangential mapping operator and combination property we write:

$$\vec{Z}(t) = d\overline{\overline{T}}^{t}_{\vec{X}_0}.\vec{Z}_0,$$
$$d\overline{\overline{T}}^{t+\tau}_{\vec{X}_0}.\vec{Z}_0 = d\overline{\overline{T}}^{t+\tau}_{\vec{X}_\tau}.(d\overline{\overline{T}}^{\tau}_{\vec{X}_0}\vec{Z}_0),$$
$$\left\|\vec{Z}_0\right\| = 1.$$

The symbol $d\overline{\overline{T}}^{t}_{\vec{X}_0}$ is representing all of the facts about the map from $\vec{Z}_0$ to $\vec{Z}(t)$, in particular showing that it is related to the nonlinear solution through the point $\vec{X}_0$. It is called the tangent map. This mapping clearly has the composition property where τ is any arbitrary time. That is, the solution after time $t+\tau$ is the same as propagating the initial condition forward for a time t using the new initial condition for the nonlinear solution $\vec{X}(t)$:

$$d\overline{\overline{T}}^{t+\tau}_{\vec{X}_0}.\vec{Z}_0 = d\overline{\overline{T}}^{t+\tau}_{\vec{X}_\tau}.(d\overline{\overline{T}}^{\tau}_{\vec{X}_0}\vec{Z}_0).$$

Now assume that, beginning with $\left\|\vec{Z}_0\right\| = 1$, the magnitude of the vector $\vec{Z}(t)$ gets larger than we want to use in a computer, when $t=\tau$. Let us designate its magnitude by $\left\|\vec{Z}\ (\tau)\right\| = \alpha_1$. Instead of continuing to use this large solution we consider a new solution which has unit magnitude, Then the unit vector is $\vec{Z}_1^{\,0}(\tau) = \dfrac{\vec{Z}(\tau)}{\left\|\vec{Z}(\tau)\right\|} = \dfrac{\vec{Z}(\tau)}{\alpha_1}$, which is a new initial condition for the variational

equation. The solution of the dynamic equation at $t = \tau$ is: $\vec{X}_\tau(t) = \vec{X}_\tau(t = \tau, \vec{X}_0)$. When n=2, the new solution for the variational equation is $\vec{Z}_1(\tau)$ and has magnitude $\left\|\vec{Z}(\tau)\right\| = \alpha_1$,

$\vec{Z}_2^0(\tau) = \dfrac{\vec{Z}_1(\tau)}{\left\|\vec{Z}_1(\tau)\right\|} = \dfrac{\vec{Z}_1(\tau)}{\alpha_2}$. According to the combinational property and linearity of the variational equations, the magnitude of $\left\|\vec{Z}(2\tau)\right\|$ is the product of α_1 and α_2: $\left\|\vec{Z}(2\tau)\right\| = \alpha_1\alpha_2$. We can proceed in this fashion, taking new initial conditions every τ time. Substituting this into the definition for the Lyapunov exponent at n=k yields, $\vec{Z}_k^0(\tau) = \dfrac{\vec{Z}_{k-1}(\tau)}{\left\|\vec{Z}_{k-1}(\tau)\right\|} = \dfrac{\vec{Z}_1(\tau)}{\alpha_k}$, where $\alpha_k = \left\|\vec{Z}_{k-1}(\tau)\right\|$. The magnitude of variational equation at $t = n\tau$ is $\left\|\vec{Z}(n\tau)\right\| = \alpha_1\alpha_2...\alpha_{n-1}\alpha_n$ and the maximum Lyapunov exponent is given by:

$$\begin{aligned}
\lambda_{\max}(\vec{X}_0) &= Lim_{t\to\infty}\left(\frac{1}{t}\right)\log_2\left(\frac{\left\|\vec{Z}(t)\right\|}{\left\|\vec{Z}_0\right\|}\right) \\
&= Lim_{n\to\infty}\left\{\frac{\log_2\left(\alpha_1\alpha_2...\alpha_{n-1}\alpha_n\right)}{n\tau}\right\} \\
&= Lim_{n\to\infty}\left\{\frac{1}{n\tau}\right\}\sum_{k=1}^{n}\log_2\left(\alpha_k\right) \;\; bits/iteration .
\end{aligned}$$

Hence it is necessary for $\lambda_{\max}(\vec{X}_0) > 0$ if the system is sensitive to initial conditions in the region $\vec{X}_0$. Obtaining the numerical solution of $\vec{Z}(t)$ usually requires the simultaneous numerical solution of the nonlinear equation for $\vec{X}(t)$. Thus, the iterates $\vec{X}(n\varepsilon)$ must be substituted into the coefficients of the variational equations $\overline{\overline{J}}\Big|_{\vec{X}(n\varepsilon)}$, before they can be iterated to give the values $\vec{Z}\left((n+1)\varepsilon\right)$. It might also be noted that τ is frequently taken to be between one and five times the integration number used in the numerical computations. Numerical evidence will be provided

when nonlinear behavior in a single Josephson junction and the Lorenz model system is investigated.

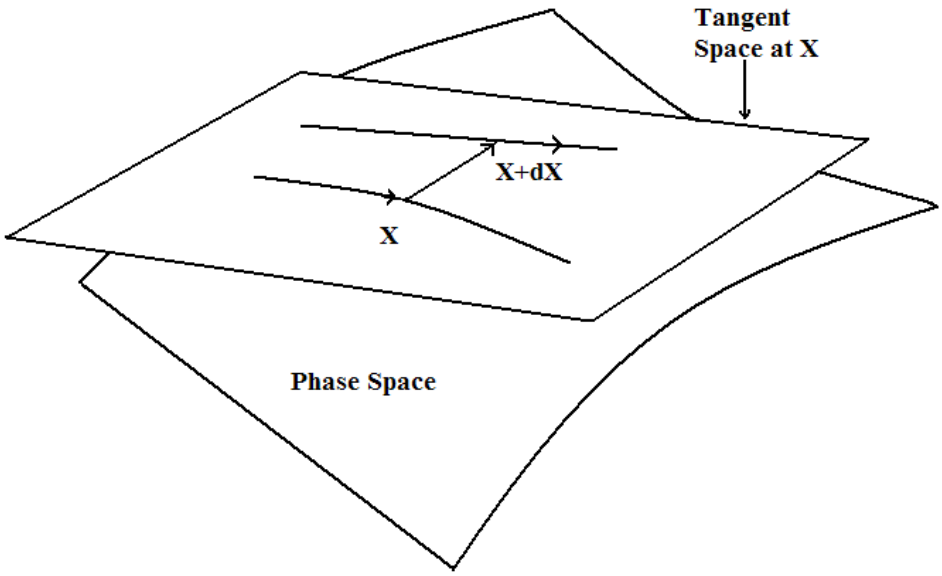
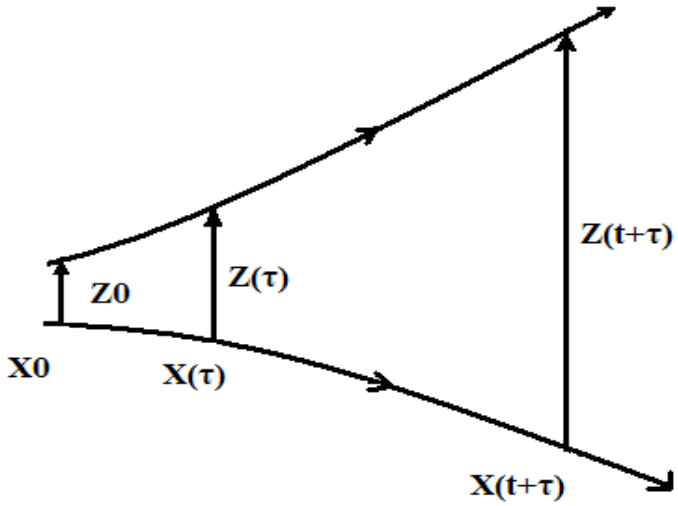
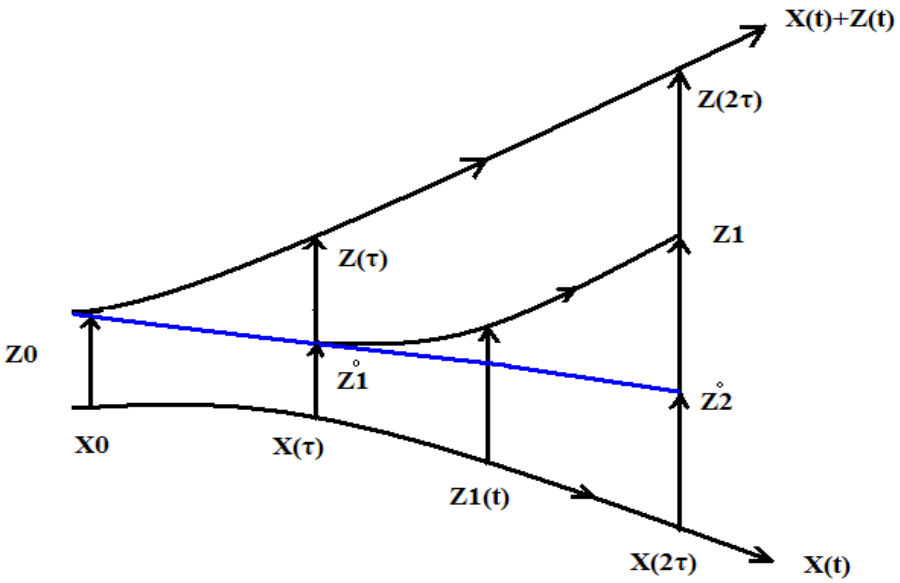


Figure 2.4 Tangent space trajectories



(a)

Figure 2.5 (a) Divergence of trajectories.



(b)

Figure 2.5, cont. (b) Reorthonormalization at time increment τ.

A subscript 1 is appended to this exponent, because it is only one of N exponents for an N-dimensional system. It is, however, one of the most important exponents, because none of the others are larger in value. Consequently, chaos is possible if and only if the sign of at least one exponent is positive. Due to the contraction along the second principal direction, the Lyapunov exponent λ_2 is negative. If the exponents are all negative, the system has a stable limit point or cycle only. Chaos cannot develop in such a system. The third Lyapunov exponent is zero because there is no deformation along the trajectory itself, as the Lyapunov exponents are invariant to linear transformations such as shifts and rotations. With these three Lyapunov exponents, we could write their signs in the following shorthand notation: (+, 0, −) which specifies chaos for flows. In general, the j^{th} Lyapunov exponent λ_j is defined as the growth rate of the j^{th} principal direction but one λ_j will be equal to zero due to the non-contracting reference trajectory. The existence of the above limit is not guaranteed for many physical dynamical systems such as the fat baker's map, particularly when contaminated by external noise [28].

2.4.1 Calculation of Second Lyapunov Exponent

To measure the next most rapid growth rate, we can use the fact that most initial vectors will move into the plane subtended by the two principal axes with the largest growth rates. If we can measure how the area in this plane changes with time then, since area is proportional to $R_1 2^{\lambda_1 t} + R_2 2^{\lambda_2 t}$, $\lambda_1 + \lambda_2 = \lim_{t\to\infty} \frac{1}{t} \log_2 (area)$ $[bits/s]$. If we know λ_1 by our previous calculation, then we can obtain λ_2 from this result. The difficulty which now occurs is that it takes two vectors to obtain the behavior of the area, but these two vectors tend to become parallel along the dominating principal axis. Thus there is a second computational difficulty, caused by the angle between two vectors becoming too small to compute. A nice way to overcome this difficulty is to orthogonalize the vectors, using a Gram-Schmidt method, at the same time as we renormalize the vectors to overcome the magnitude problem. The procedure is not difficult to visualize, although it is a little laborious to write down in detail. Consider two vector solutions $\vec{Z}^{(1)}(t), \vec{Z}^{(2)}(t)$, which tend to become nearly parallel at $t=\tau$. Let the magnitude of $\alpha_1^{(1)} = \left\| \vec{Z}^{(1)}(\tau) \right\|$ so that $\vec{Z}_1^{(1)}(\tau) = \frac{\vec{Z}^{(1)}(\tau)}{\left\| \vec{Z}^{(1)}(\tau) \right\|} = \frac{\vec{Z}^{(1)}(\tau)}{\alpha_1^{(1)}}$ is a unit vector. The area associated with the two vectors $\vec{Z}^{(1)}(t), \vec{Z}^{(2)}(t)$ at $t=\tau$ equals the base times the height of the parallelogram created by two vectors $\vec{Z}^{(1)}(t), \vec{Z}^{(2)}(t)$, that is:

$$Area(t=\tau) = \left\| \vec{Z}^{(1)}(\tau) \right\| \cdot \left\| \{ \vec{Z}^{(2)}(\tau) - \left\langle \vec{Z}^{(2)}(\tau) \middle| \vec{Z}_1^{(1)}(\tau) \right\rangle \vec{Z}_1^{(1)}(\tau) \} \right\|,$$

where bracket is the scalar product of the two vectors. If we now define a second new vector which is orthogonal to $\vec{Z}_1^{(1)}(\tau)$ and which is a unit vector if:

$$\alpha_1^{(2)} = \left\| \{ \vec{Z}^{(2)}(\tau) - \left\langle \vec{Z}^{(2)}(\tau) \middle| \vec{Z}_1^{(1)}(\tau) \right\rangle \vec{Z}_1^{(1)}(\tau) \} \right\|.$$

First unit vector and the area associated with the two vectors are:

$$\vec{Z}_1^{(1)}(\tau) = \frac{\vec{Z}^{(1)}(\tau)}{\left\| \vec{Z}^{(1)}(\tau) \right\|} = \frac{\vec{Z}^{(1)}(\tau)}{\alpha_1^{(1)}},$$

$$Area(t = \tau) = \left\| \vec{Z}^{(1)}(\tau) \right\| . \left\| \{ \vec{Z}^{(2)}(\tau) - \left\langle \vec{Z}^{(2)}(\tau) \middle| \vec{Z}_1^{(1)}(\tau) \right\rangle \vec{Z}_1^{(1)}(\tau) \} \right\|.$$

Hence, the second unit vector perpendicular to $\vec{Z}_1^{(1)}(\tau)$ and the area at $t = \tau$ are:

$$\vec{Z}_1^{(2)}(\tau) = \frac{\{ \vec{Z}^{(2)}(\tau) - \left\langle \vec{Z}^{(2)}(\tau) \middle| \vec{Z}_1^{(1)}(\tau) \right\rangle \vec{Z}_1^{(1)}(\tau) \}}{\alpha_1^{(2)}}$$

$$Area(t = \tau) = \left\| \vec{Z}^{(1)}(\tau) \right\| . \left\| \{ \vec{Z}^{(2)}(\tau) - \left\langle \vec{Z}^{(2)}(\tau) \middle| \vec{Z}_1^{(1)}(\tau) \right\rangle \vec{Z}_1^{(1)}(\tau) \} \right\| = \alpha_1^{(1)} \alpha_1^{(2)}$$

Use $\vec{Z}_1^{(1)}(\tau)$ and $\vec{Z}_1^{(2)}(\tau)$ as initial unit vectors for the variational equation:

$$\vec{Z}_2^{(1)} = \frac{\vec{Z}_1^{(1)}}{\left\| \vec{Z}_1^{(1)} \right\|},$$

$$\vec{Z}_2^{(2)}(2\tau) = \{ \vec{Z}_1^{(2)}(2\tau) - \left\langle \vec{Z}_1^{(2)}(2\tau) \middle| \vec{Z}_2^{(1)}(2\tau) \right\rangle \vec{Z}_2^{(1)}(2\tau) \},$$

$$\alpha_2^{(1)} = \left\| \vec{Z}_1^{(1)} \right\|,$$

$$\alpha_2^{(2)} = \left\| \{ \vec{Z}_1^{(2)}(2\tau) - \left\langle \vec{Z}_1^{(2)}(2\tau) \middle| \vec{Z}_2^{(1)}(2\tau) \right\rangle \vec{Z}_2^{(1)}(2\tau) \} \right\|,$$

$$Area\ (t = 2\tau) = \{ \alpha_1^{(1)}(\tau) \alpha_1^{(2)}(\tau) \} \{ \alpha_2^{(1)}(2\tau) \alpha_2^{(2)}(2\tau) \},$$

$$Area(t = n\tau) = \{ \alpha_1^{(1)}(\tau) \alpha_1^{(2)}(\tau) \} \{ \alpha_2^{(1)}(2\tau) \alpha_2^{(2)}(2\tau)) \} \ldots \{ \alpha_n^{(1)}(n\tau) \alpha_n^{(2)}(n\tau) \},$$

$$\lambda_1 + \lambda_2 = Lim_{N \to \infty} \left\{ \frac{\sum_{k=1}^{N} \log_2 (Area(t = n\tau))}{N\tau} \right\},$$

$$\lambda_1 = Lim_{N \to \infty} \left\{ \frac{\sum_{k=1}^{N} \log_2 (\alpha_k^{(1)})}{N\tau} \right\},$$

$$\lambda_2 = Lim_{N\to\infty}\left\{\frac{\sum_{k=1}^{N}\log_2(\alpha_k^{(2)})}{N\tau}\right\}.$$

2.4.2 Lyapunov Spectrum in Three and Four Dimensions

The mathematical formulation of the Lyapunov spectrum computation is described below. We start from the nonlinear dynamical system equations and develop the variational equations. Examples in 3-dimensional systems including Josephson junction system, the Lorenz model and the Josephson Tetrode microbridge are provided in chapters 3 through 5.

$$\frac{d\vec{X}}{dt} = \vec{F}(\vec{X}, \vec{X}_0, \vec{C}),$$
$$\frac{dZ^{(i)}(t)}{dt} = J.Z^{(i)}(t),$$

$$\left\|\vec{Z}_0^{(i)}\right\| = 1, \ \ i = 1,2,3,...$$
$$\vec{Z}_1^{(1)}(\tau) = \frac{\vec{Z}^{(1)}(\tau)}{\alpha_1^{(1)}},$$
$$\vec{Z}_1^{(2)}(\tau) = \{\vec{Z}^{(2)}(\tau) - \left\langle \vec{Z}^{(2)}(\tau)\middle|\vec{Z}_1^{(1)}(\tau)\right\rangle \vec{Z}_1^{(1)}(\tau)\} \ \ /\alpha_1^{(2)},$$

$$\vec{Z}_1^{(3)}(\tau) = \{\vec{Z}^{(3)}(\tau) - \left\langle \vec{Z}^{(3)}(\tau)\middle|\vec{Z}_1^{(2)}(\tau)\right\rangle \vec{Z}_1^{(2)}(\tau) - \left\langle \vec{Z}^{(3)}(\tau)\middle|\vec{Z}_1^{(1)}(\tau)\right\rangle \vec{Z}_1^{(1)}(\tau)\} \ \ /\alpha_1^{(3)},$$

$$Volume(t=\tau) = \{\alpha_1^{(1)}(\tau)\alpha_1^{(2)}(\tau)\alpha_1^{(3)}\},$$
$$Volume(t=n\tau) = \{\alpha_1^{(1)}(\tau)\alpha_1^{(2)}(\tau)\alpha_1^{(3)}(\tau)\}\times\{\alpha_2^{(1)}(2\tau)\alpha_2^{(2)}(2\tau)\alpha_3^{(3)}(2\tau)\}\times\ldots$$
$$\times...\{\alpha_n^{(1)}(n\tau)\alpha_n^{(2)}(n\tau)\alpha_n^{(3)}(n\tau)\},$$

$$\lambda_i = Lim_{N\to\infty}\{\frac{\sum_{k=1}^{N}\log_2\alpha_k^{(i)}}{N\tau}\}, \ \ i = 1,2,3.$$

The Lyapunov exponents spectrum in 4-dimensional systems like a feedback-controlled hyperchaotic system and coupled Duffing-Van der Pol oscillators are calculated as follows:

$$\left\|\vec{Z}_0^{(i)}\right\|=1,\ \ i=1,2,3,4,$$

$$\vec{Z}_1^{(1)}(\tau)=\frac{\vec{Z}^{(1)}(\tau)}{\alpha_1^{(1)}},$$

$$\vec{Z}_1^{(2)}(\tau)=\left(\frac{1}{\alpha_1^{(2)}}\right)\left\{\vec{Z}^{(2)}(\tau)-\left\langle\vec{Z}^{(2)}(\tau)\middle|\vec{Z}_1^{(1)}(\tau)\right\rangle\vec{Z}_1^{(1)}(\tau)\right\},$$

$$\vec{Z}_1^{(3)}(\tau)=\left(\frac{1}{\alpha_1^{(3)}}\right)\left\{\vec{Z}^{(3)}(\tau)-\left\langle\vec{Z}^{(3)}(\tau)\middle|\vec{Z}_1^{(2)}(\tau)\right\rangle\vec{Z}_1^{(2)}(\tau)-\left\langle\vec{Z}^{(3)}(\tau)\middle|\vec{Z}_1^{(1)}(\tau)\right\rangle\vec{Z}_1^{(1)}(\tau)\right\},$$

$$\vec{Z}_1^{(4)}(\tau)=\left(\frac{1}{\alpha_1^{(4)}}\right)\left\{\begin{array}{l}\vec{Z}^{(4)}(\tau)-\left\langle\vec{Z}^{(4)}(\tau)\middle|\vec{Z}_1^{(3)}(\tau)\right\rangle\vec{Z}_1^{(3)}(\tau)-\left\langle\vec{Z}^{(4)}(\tau)\middle|\vec{Z}_1^{(2)}(\tau)\right\rangle\vec{Z}_1^{(2)}(\tau)-\\ -\left\langle\vec{Z}^{(4)}(\tau)\middle|\vec{Z}_1^{(1)}(\tau)\right\rangle\vec{Z}_1^{(1)}(\tau)\end{array}\right\},$$

$$Volume(t=\tau)=\left\{\alpha_1^{(1)}(\tau)\alpha_1^{(2)}(\tau)\alpha_1^{(3)}(\tau)\alpha_1^{(4)}(\tau)\right\},$$

$$Volume(t=n\tau)=\left\{\alpha_1^{(1)}(\tau)\alpha_1^{(2)}(\tau)\alpha_1^{(3)}(\tau)\alpha_1^{(4)}(\tau)...\alpha_n^{(1)}(n\tau)\alpha_n^{(2)}(n\tau)\alpha_n^{(3)}(n\tau)\alpha_n^{(4)}(n\tau)\right\},$$

$$\lambda_i=Lim_{N\to\infty}\{\frac{\sum_{k=1}^{N}\log_2\alpha_k^{(i)}}{N\tau}\},\ \ i=1,2,3,4.$$

In dissipative dynamical systems, the volume of initial conditions in phase space is decreased under dynamical evolution $\frac{dV}{dt}=\iiint\sum_{i=1}^{N}(\frac{\partial\vec{F}}{\partial\vec{X}})d\vec{X},\vec{X}\in\Re^N$; which has negative value if $\frac{dV}{dt}<0$ for a dissipative system and $\frac{dV}{dt}=0$ for Hamiltonian systems where $V(t)\approx 2^{(\sum_i\lambda_i t)}$ or $\nabla.\vec{F}=\sum_i\lambda_i$.

2.5 *N*-Dimensional Maps

The previous approaches with some differences, also apply to discrete maps. A trajectory of a flow in a phase space has a direction of evolution in time *t*. Thus, the time elapsed between two distinct points on the trajectory must be positive. If we take two distinct points on the same trajectory that are separated by a fixed interval, then the difference between the points cannot

diverge or contract. Consequently, since the Lyapunov exponent along the trajectory itself must always be zero, flows must have at least one zero exponent. On the other hand, if the attractor is generated by a map, or is reconstructed from time series experimental data, the sense of the time arrow is lost in state space, thus leading to no zero Lyapunov exponents. Consider the *N*-dimensional map

$$\vec{X}_{n+1} = \vec{F}(\vec{X}_n, \vec{X}_0, \vec{C}),$$
$$\delta\vec{X}_{n+1} = \frac{\partial\vec{F}}{\partial\vec{X}}\Big|_{\vec{X}_n} \delta\vec{X}_n = \overline{\overline{J}}\Big|_{\vec{X}_n} .\delta\vec{X}_n = \overline{\overline{J}}\Big|_{\vec{X}_n} .\vec{Z}_n,$$
$$\delta\vec{X}_n \equiv \vec{Z}_n.$$

Then the variational equation is written as $\vec{Z}_{n+1} = \overline{\overline{J}}\Big|_{\vec{X}_n} .\vec{Z}_n$, with initial unit vector, $\vec{Z}_0$. The solution of variational equation at step *n* will be

$$\vec{Z}_{n+1} = \overline{\overline{J}}\Big|_{\vec{X}_n} .\vec{Z}_n = \overline{\overline{J}}\Big|_{\vec{X}_n} .\overline{\overline{J}}\Big|_{\vec{X}_{n-1}} .\overline{\overline{J}}\Big|_{\vec{X}_{n-2}} ...\overline{\overline{J}}\Big|_{\vec{X}_1} .\overline{\overline{J}}\Big|_{\vec{X}_0} .\vec{Z}_0$$

$\overline{\overline{J}}\Big|_{\vec{X}_n}$,..., $\overline{\overline{J}}\Big|_{\vec{X}_1}$, $\overline{\overline{J}}\Big|_{\vec{X}_0}$ are Jacobian matrices determined for each point like $\vec{X}_m$.

By the Lyapunov exponent definition we can write:

$$\lambda_{\max}(\vec{X}_0) = Lim_{n\to\infty}\left(\frac{1}{n}\right)\log_2\left(\frac{\|\vec{Z}_n\|}{\|\vec{Z}_0\|}\right).$$

This relation formulates the discrete map Lyapunov exponent estimation.

2.6 Lyapunov Exponents Spectrum

We have seen that the signs of the Lyapunov exponents alone are sufficient to determine the stability of a dynamical system. It is convenient to arrange all the exponents starting from the largest positive on the left, representing the maximum stretching direction, to the largest negative

on the right, representing the largest contraction direction and then substitute this arrangement with the corresponding signs of the components as this constitutes the Lyapunov exponent spectrum and the symbolic Lyapunov spectrum, respectively. The latter is often used in describing system stability completely.

A one dimensional map has a single Lyapunov exponent. If it is negative, the map has either a limit-point stability or a limit-cycle stability. For *N*-Dim Hamiltonian or non-Hamiltonian dissipative maps, the stretching and contraction along all the *N* principal directions in phase space produce *N* Lyapunov exponents, usually arranged in the symbolic Lyapunov spectrum in order to facilitate stability analysis. If the exponents are equal but have opposite signs, the expansion in one direction is offset by the contraction in the other direction, thus preserving the area of the hyper-ellipsoid in *N*-Dim. On the other hand, in non-Hamiltonian (dissipative) systems, the area-preserving principle does not apply. In fact, the sum of all the *N* Lyapunov exponents must be negative for physical dissipative dynamical system. Consequently, a physical dynamical system with (+, +, 0) cannot exist. Chaotic dynamical systems described by differential equations that evolve in continuous time dynamic flows must have one positive and one zero Lyapunov exponent. The third negative exponent must also exist so that the sum of the three exponents is negative. This requirement also implies that at least three dimensions are necessary for chaos to develop in such differential equations. On the other hand, for iterated maps, no zero exponent is required. Consequently, for 2-Dim maps, just two exponents (+, −) are sufficient to produce a negative sum in order for the phase space volume to contract on the average. Stable periodic attractors have only zero and negative Lyapunov exponents. For example in 3-Dim, we have either (0, −, −) where the zero corresponds to the limit-cycle trajectory itself or (0, 0, −) for an attracting 2-torus. Chaotic attractors have only one finite

positive Lyapunov exponent. For example in 3-D, there is only one symbolic Lyapunov spectrum of the form (+, 0, −), where the zero corresponds to the chaotic trajectory itself, with some trajectories expanding, while others are contracting. In a higher dimensional dynamical system, more than one exponent may be positive. This situation is called hyperchaos. In four dimensions there are three distinct strange attractors: (+, +, 0, −) hyperchaos, (+, 0, −, −) chaos, (+, 0, 0, −) torus chaos. Hyperchaotic systems are studied in chapter five.

2.7 Lyapunov Spectrum Estimation without Rescaling and Reorthogonalization

In this section we present another method to compute Lyapunov exponents utilizing representations of orthogonal matrices applied to decompositions of the tangent map. This method uses a minimal set of variables, and does not require renormalization or reorthogonalization (NRNO). It can be used efficiently to compute partial Lyapunov spectra. The NRNO method is based on exact differential equations for the Lyapunov exponents. Consequently the global invariances of the Lyapunov spectrum can be preserved. The key feature of this method is the use of group theoretical representations of orthogonal matrices. This results in a set of coupled ordinary differential equations for the Lyapunov exponents along with the various angles parametrising the orthogonal matrices. The system of differential equations is treated as an initial value problem and solved numerically to obtain the Lyapunov exponents. A consequence of this method is the separation between the exponents and the angles pieces in the evolution equations. To begin, we consider an *n* dimensional continuous time dynamical system:

$$\frac{d\vec{X}}{dt} = \vec{F}(\vec{X}, \vec{X}_0, \vec{C}), \vec{X} \in \Re^N.$$

Linearizing of the dynamical systems equation around this trajectory, we obtain,

$$\frac{d\vec{Z}(t)}{dt} = J\Big|_{\vec{X}_0}.\vec{Z}(t).$$

$\delta\vec{X} = \vec{X}(t) - \vec{X}(0)$ or $\delta\vec{X} = \vec{Z}(t) = \left(Z_1(t), Z_2(t), .. Z_n(t)\right)$, with $\vec{Z}(t)$ denoting deviations from the fiducial trajectory $\vec{X}(0)$ in a n-dimensional vector field $\vec{F}(\vec{X}, \vec{X}_0, \vec{C})$, and J is the Jacobian matrix. Integrating the linearized equations along the fiducial trajectory yields the tangent map Π, which takes the initial variables Z in into the time-evolved variables $\vec{Z}(t) = \Pi.\vec{Z}_0(t)$. Ξ is a $N \times N$ matrix given by $\Xi = Lim_{t\to\infty}\{\frac{\tilde{\Pi}\Pi}{2t}\}$, and $\tilde{\Pi}$ denotes the matrix transpose of Π. The Lyapunov exponents then equal the logarithm of the eigenvalues of Ξ. Π is important in the evaluation of Lyapunov exponents and its evolution is determined by: $\frac{d\Pi(t)}{dt} = J.\Pi(t)$. The evolution equations are well-behaved if the matrix Λ is defined as: $\Lambda(t) = \Pi(t).\tilde{\Pi}(t)$. Substitution on evolution equation for Λ yields: $\frac{d\Lambda(t)}{dt} = J.\Lambda(t) + \Lambda(t).\tilde{J}$. The matrix Λ is symmetric and positive-definite. It can be written as an exponential of a symmetric matrix $\Lambda(t) = e^{\mathrm{P}(t)}$ where P(t) is also a symmetric matrix. Hence P(t) can be diagonalized using orthogonal matrices $\Phi\Theta\Phi^{-1}$ that is, $\mathrm{P}(t) = \Phi\Theta\Phi^{-1}$. Thus Λ($t$) can be written as $\Lambda(t) = e^{\Phi\Theta\Phi^{-1}}$, where Φ is an $N \times N$ orthogonal matrix, and Θ is an $N \times N$ diagonal matrix. Also the relation $\Phi^{-1} = \tilde{\Phi}$ holds. From standard properties of matrix exponentials, it follows that $\Lambda(t) = \Phi e^{\Theta}\Phi^{-1}$. The rescaling is not required since the diagonal matrix Θ is already in the exponent (the diagonal elements are just the Lyapunov exponents multiplied by time). To proceed further, the matrix Π is written as the product $\Pi = \Omega\mathrm{H}$ of an orthogonal $N \times N$ matrix Ω and an upper-triangular $N \times N$

matrix H with positive diagonal entries. Substituting this into above evolution equation, we obtain: $\dot{\Omega}\mathrm{H} + \Omega\dot{H} = J\Omega H$. Multiplying the above equation by $\tilde{\Omega}$ from the left and H^{-1} from the right, we get $\tilde{\Omega}\dot{\Omega} + \dot{H}H^{-1} = \tilde{\Omega}J\Omega$. Note that $\tilde{\Omega}\dot{\Omega}$ is a skew matrix for any orthogonal matrix Ω and $\dot{H}H^{-1}$ is still an upper triangular matrix. As before, we now employ an explicit representation of the orthogonal matrix Ω representing it as a product of $N(N\text{-}1)/2$ orthogonal matrices, each of which corresponds to a simple rotation in the i-j^{th} plane ($i < j$). We denote the matrix corresponding to this rotation by $\Psi^{(i,j)}$; its matrix elements are given by:

$$\begin{aligned}
&\Psi^{(i,j)}_{k,l} = 1, \;\; k = l \neq i, j, \\
&\Psi^{(i,j)}_{k,l} = \cos\alpha, \;\; k = l = i, \; j, \\
&\Psi^{(i,j)}_{k,l} = \sin\alpha, \;\; k = i, \;\; l = j, \\
&\Psi^{(i,j)}_{k,l} = -\sin\alpha, \;\; k = j, \;\; l = i, \\
&\Psi^{(i,j)}_{k,l} = 0, \;\; \textit{otherwise.}
\end{aligned}$$

Here α denotes an angle variable. Thus, the $N \times N$ matrix Ω is represented by:

$$\Omega = \Psi^{(1,2)}\Psi^{(1,3)}\Psi^{(1,4)}\Psi^{(1,n)}\ldots\Psi^{(2,3)}\Psi^{(2,4)}\ldots\Psi^{(n-1,n)}.$$

Hence Ω is parametrized by $N(N\text{-}1)/2$ angles which are denoted by α_i (i = 1, 2,…, $N(N\text{-}1)/2$).

$$H = \begin{pmatrix} e^{\lambda_1} & h_{12} & \cdots & \cdots & h_{1N} \\ 0 & e^{\lambda 2} & h_{23} & \cdots & h_{2N} \\ 0 & 0 & e^{\lambda 3} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & e^{\lambda_N} \end{pmatrix}.$$

The upper-triangular matrix H has positive diagonal entries. The quantities h_{ij} represent the upper-diagonal terms in H. The quantities h_{ij} are of no concern since they are not present in the final equations. Substituting the above two expressions we obtain the matrix $\mathrm{Y} = \tilde{\Omega}J\Omega$ and comparing diagonal elements on both sides we have:

$$\tilde{\Omega}\dot{\Omega} = \begin{pmatrix} 0 & -\omega_1(\dot{\alpha}) & -\omega_2(\dot{\alpha}) & \cdots & -\omega_{N-1}(\dot{\alpha}) \\ \omega_1(\dot{\alpha}) & 0 & \cdots & \cdots & -\omega_{2N-3}(\dot{\alpha}) \\ \omega_2(\dot{\alpha}) & 0 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_{N-1}(\dot{\alpha}) & \cdots & \cdots & \omega_{\frac{N(N-1)}{2}}(\dot{\alpha}) & 0 \end{pmatrix},$$

$$\dot{H}H^{-1} = \begin{pmatrix} \dot{\lambda}_1 & h'_{12} & h'_{13} & \cdots & h'_{1N} \\ 0 & \dot{\lambda}_2 & h'_{23} & \cdots & h'_{2N} \\ 0 & 0 & \dot{\lambda}_3 & \cdots & h'_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dot{\lambda}_N \end{pmatrix}.$$

The Lyapunov exponents are equal to λ_i/t in the limit $t \to \infty$. Thus, the Lyapunov exponents can be obtained by solving the above differential equations for long times. However, since the right hand side depends on the angles α_i, we also require differential equations governing the evolution of these angles.

$$\Delta \equiv \tilde{\Omega} J \Omega = \begin{pmatrix} \dot{\lambda}_1 & h''_{12} & h''_{13} & \cdots & h''_{1n} \\ \omega_1(\dot{\alpha}) & \dot{\lambda}_2 & h''_{23} & \cdots & h''_{2n} \\ \omega_2(\dot{\alpha}) & 0 & \dot{\lambda}_3 & \cdots & h''_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_{N-1}(\dot{\alpha}) & 0 & 0 & \omega_{\frac{N(N-1)}{2}}(\dot{\alpha}) & \dot{\lambda}_N \end{pmatrix}$$

Differential equations for the angles can be obtained by comparing the sub-diagonal elements. This gives:

$$\begin{aligned} &\omega_1(\dot{\alpha}) = \Delta_{21}, \\ &\omega_2(\dot{\alpha}) = \Delta_{31}, \\ &\vdots \\ &\omega_{\frac{N(N-1)}{2}}(\dot{\alpha}) = \Delta_{N,N-1}. \end{aligned}$$

These are $N(N\text{-}1)/2$ ordinary differential equations that can be solved to obtain the Lyapunov exponents. This system of differential equations has the feature that the equation for λ_1 depends only on the first (N-1) α_i's. Therefore, if one is interested in only the largest Lyapunov exponent, one needs to solve only N equations (as opposed to $N(N\text{-}1)/2$ for the full spectrum). The equation for λ_2 depends only on the first ($2N$-3) α_i's. Therefore, to obtain the first two Lyapunov exponents, one needs to solve only $2N-1$ equations. In general, to solve for the first K Lyapunov exponents, one has to solve $K(2N\text{-}K\text{+}1)/2$ equations which is always less than $N(N\text{-}1)/2$, the total number of equations for $K < N$. This is in contrast to the situation for the conventional continuous QR or SVD methods where it is computationally costlier to evaluate a partial spectrum. One advantage is that the NRNO method uses the minimal set of dynamical variables to characterize the system. Therefore a partial Lyapunov spectrum can be computed using a fewer number of equations as compared to the computation of the full spectrum. Other advantages are that numerical errors can never lead to loss of orthogonality and that this approach is straightforward to generalize to higher dimensions. On the other hand, in low dimensional systems the modified GSR method converges quickly, the exponents are in order in the spectrum and partial Lyapunov spectra can be calculated. While in the NRNO method, our experience shows that exponents mix up and appear out of order.

2.8 Henon Map

The Henon map is a prototypical two-dim invertible iterated map with chaotic solutions proposed by the French astronomer Michel Henon in 1976 as a simplified model of the Poincare map for the Lorenz model. The Henon map can illustrate the basic concepts of complex dynamical systems from non-linearity to chaos and is defined by the following quadratic (nonlinear) recursive map:

$$X_{n+1} = 1 - aX_n^2 + Y_n,$$
$$Y_{n+1} = bX_n.$$

where $0 \le a \le 2$ and $|b| < 1$. The parameter b is a measure of the rate of area contraction or dissipativity. The Henon map is the most general two-dim quadratic map with the property that the contraction is independent of X and Y. For b = 0, the Henon map reduces to the quadratic map, which is conjugate to the logistic map.

In order to compute the Lyapunov exponent for a Henon map, we need to solve the variational equation for this dynamic system. The solution of variational equation at step $n+1$ will be:

$$\vec{Z}_{n+1} = (\delta X_{n+1}, \delta Y_{n+1}) = J_n\big|_{(X,Y)} . \vec{Z}_n \ ,$$
$$\vec{Z}_n = (\delta X_n, \delta Y_n) = J_{n-1}\big|_{(X_{n-1}, Y_{n-1})} . \vec{Z}_{n-1} \ ,$$
$$= J_{n-1}\big|_{(X_{n-1}, Y_{n-1})} . J_{n-2}\big|_{(X_{n-2}, Y_{n-2})} \cdots J_1\big|_{(X_1, Y_1)} J_0\big|_{(X_0, Y_0)} \vec{Z}_0 \ .$$

$\vec{Z}_0 = (\delta X_0, \delta Y_0)$ is the initial unit vector. J_m is Jacobian matrix evaluated for each point (X_m, Y_m). The Jacobian matrices are

$$J_n\big|_{(X_n, Y_n)} = \left(\frac{\partial \vec{F}}{\partial \vec{X}} \right)_{(X_n, Y_n)} = \begin{pmatrix} -2aX_n & 1 \\ b & 0 \end{pmatrix}.$$

By the Lyapunov exponent definition we can write,

$$\lambda_1 = Lim_{n \to \infty} \log_2 \left(\frac{\|\vec{Z}_n\|}{\|\vec{Z}_0\|} \right).$$

We have investigated the existence and transformation of the chaotic attractor of the Henon mapping with numerical methods. In general, bounded solutions exist for the Henon map over a range of a and b values. Notice that for all values $|b| < 1$, and for a wide range of values of a, the mapping is contracting the area, and has a trapping region. A portion of this range yields periodic

and chaotic states. These are illustrated in the Henon attractors point density distribution for control parameter a when b=0.3. Figure 2.6 shows a 1-Cycle attractor at a = 0 followed by a 1→2 Cycle transition at a = 0.36; a 2-Cycle attractor at a = 0.5 followed by a 2→4 Cycle transition at a = 0.913 where the sequence converges towards four fixed point; 4-Cycle attractor at a = 1 and 4→8 Cycle transition at a = 1.026. If a is increased beyond a critical value a_1 = 1.026 then the values of the sequence jump periodically between eight values after a certain time of transition. Figure 2.7 shows that if a is increased further then at a critical value a = 1.0511, the period length doubles with a 8→16 Cycle transition followed by a 16→32 Cycle at a = 1.0565 and a chaotic state at a = 1.1. If a is increased further Chaos→7 Cycle transition occurs at a = 1.2266 followed by 7 Cycle attractor at a = 1.227 and another 7-cycle at a = 1.3. If a is increased further and further; then the period doubles each time with a sequence of critical values a_2, a_3. But beyond a critical value, the development becomes more and more irregular and chaotic. Some of these chaotic states are shown in Figure 2.8 for a=1.35, 1.39 and 1.4. This sequence of period doubling bifurcations and chaotic region is illustrated in Figure 2.9 for control parameter b=0.3 when parameter a varied from 0 to 1.4. The sequence of period-doubling for different values of control parameter a in 2^k to 2^{k+1} cycles and 7×2^k to $7\times2^{k+1}$ cycles are listed in tables 2.1 and 2.2. Form these values one can easily verify the Feigenbaum universal number ($\delta_F \approx 4.699$) for period-doubling cycles. This number says the distance between bifurcation points decreases by a factor of δ_F for each bifurcation.

In addition, starting from orthogonal unit vectors, (1, 0) and (0, 1) as initial values for variational equation, and averaging over 4000 iterations, we have calculated the Lyapunov exponent for Henon map for control parameter b=0.3 when parameter a varied from 1 to 1.4 (Figure 2.10). The corresponding bifurcation diagram is illustrated in Figure 2.11. λ is negative

in periodic windows, positive in chaotic regions and zero at unstable period-doubling transition points.

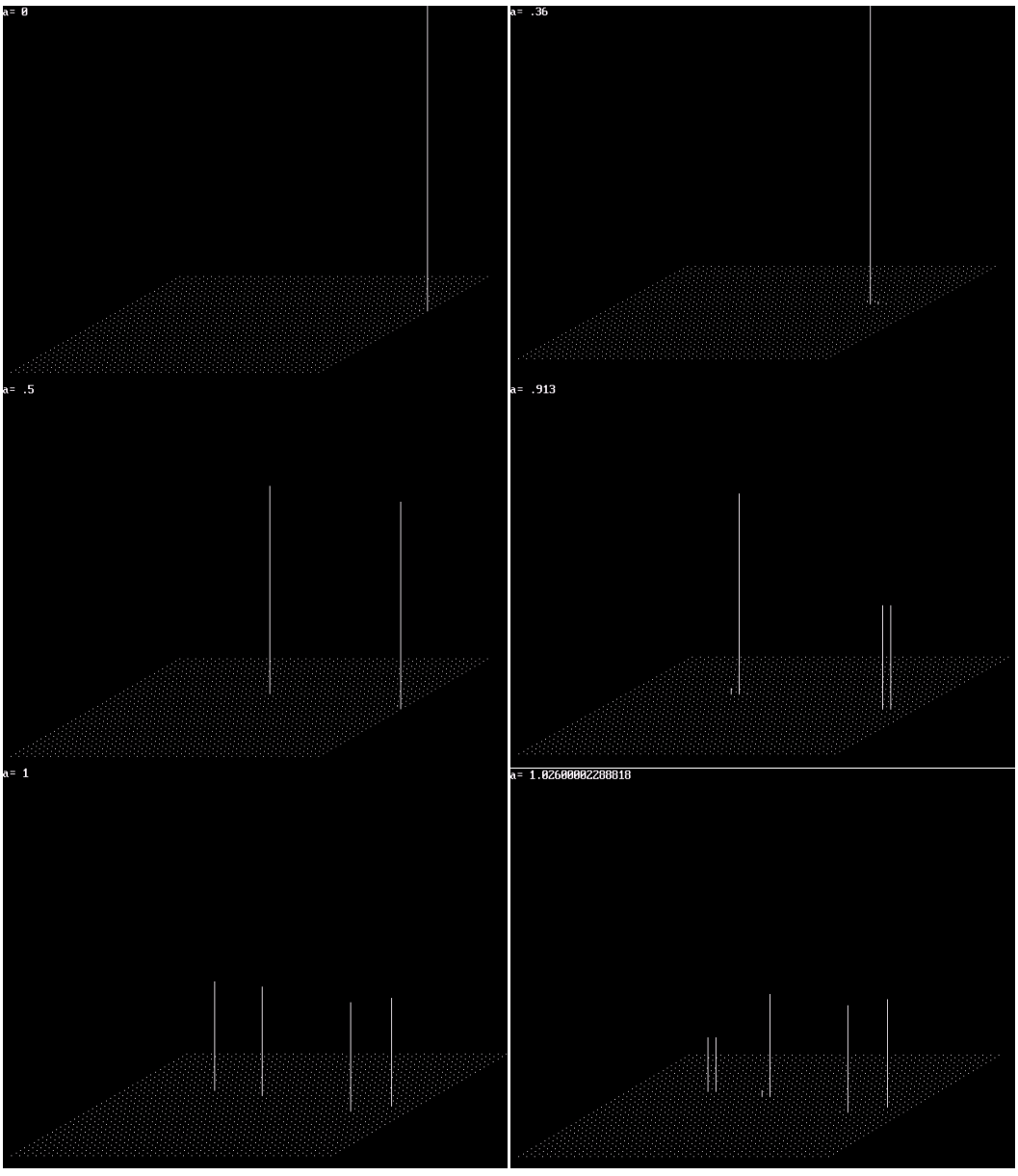


Figure 2.6 Henon map attractors point density distribution for different values of the control parameter *a* when *b*=0.3. 1-Cycle at *a* = 0, 1→2 Cycle at *a* = 0.36, 2-Cycle at *a* = 0.5, 2→4 Cycle at *a*=0.913, 4-Cycle at *a* = 1 and 4→8 Cycle at *a* = 1.026.

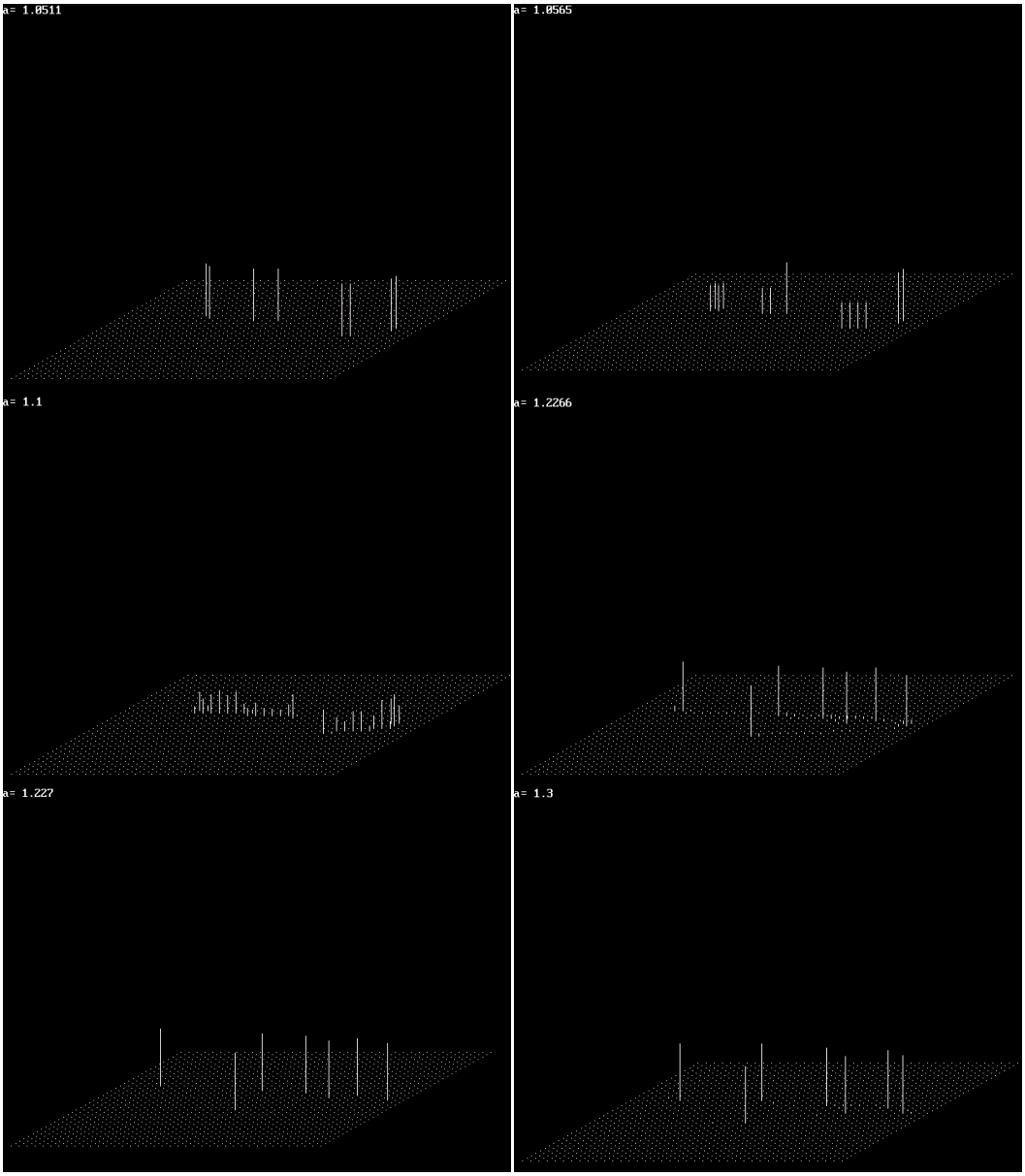


Figure 2.7 Henon map attractors density distribution for different values of the control parameter a and b=0.3. 8→16 Cycle at a = 1.0511, 16→32 Cycle at a = 1.0565, chaotic state at a = 1.1, Chaos→7 Cycle at a=1.2266, 7 Cycle at a = 1.227, other 7-cycle at a = 1.3.

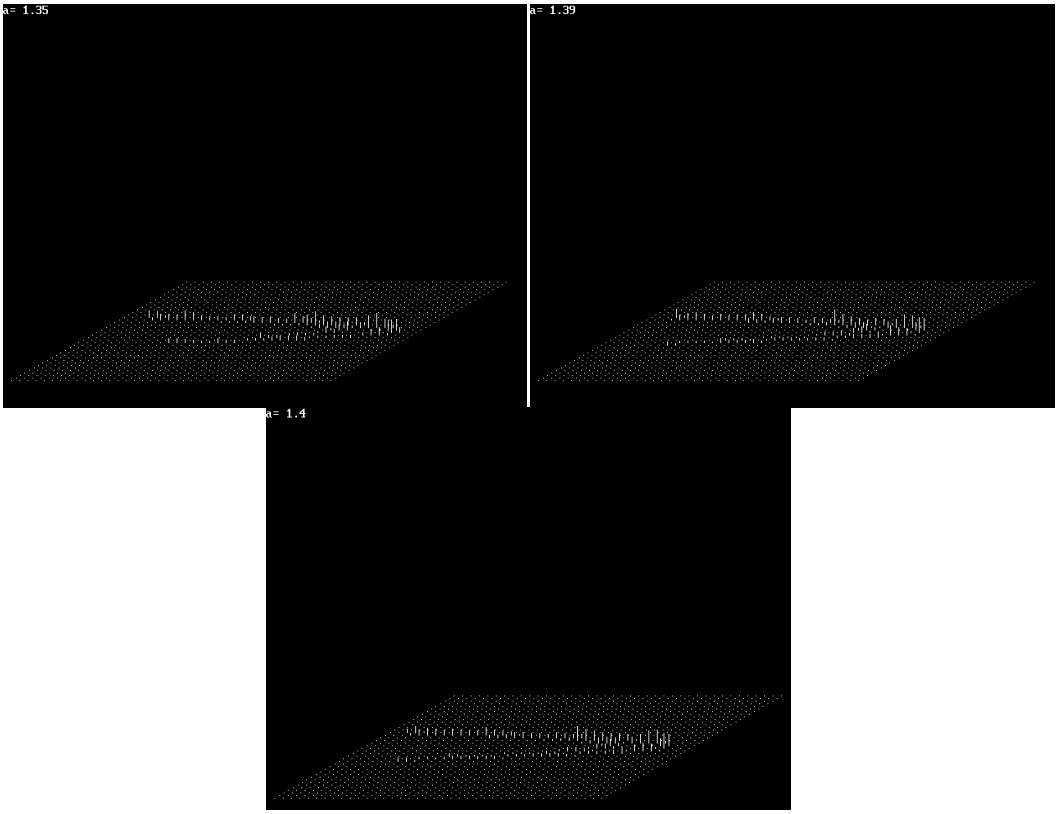


Figure 2.8 Henon map attractors density distribution for different values of the control parameter *a* and *b* = 0.3. Chaotic states at *a* = 1.35, 1.39 and 1.4.

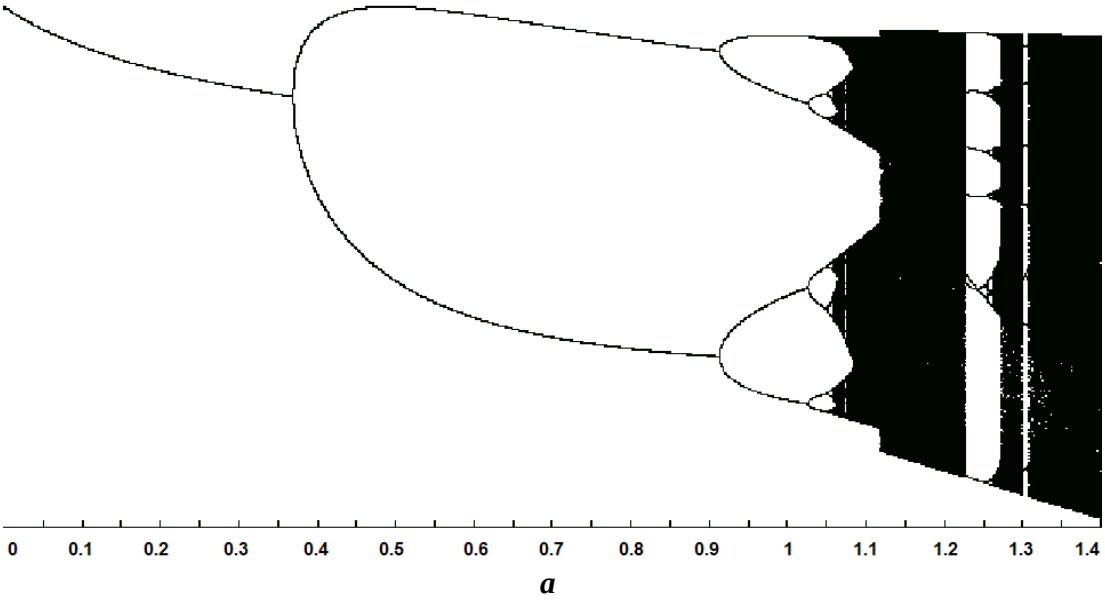


Figure 2.9 Henon map bifurcation diagram vs. control parameter *a* when *b* = 0.3.

Table 2.1 Henon bifurcation parameter values for 2^k to 2^{k+1} cycles, b = 0.3.

Henon parameter (a)	Bifurcation
0.36	1-Cycle→2-Cycle
0.913	2-Cycle→4-Cycle
1.026	4-Cycle→8-Cycle
1.0511	8-Cycle→16-Cycle
1.0565	16-Cycle→32-Cycle
1.0577	32-Cycle→64-Cycle

Table 2.2 Henon bifurcation parameter values for 7×2^k to $7\times2^{k+1}$ cycles, b = 0.3.

Henon parameter (a)	Bifurcation
1.227	7-Cycle→Chaos
1.254	7-Cycle→14-Cycle
1.260	14-Cycle→28-Cycle

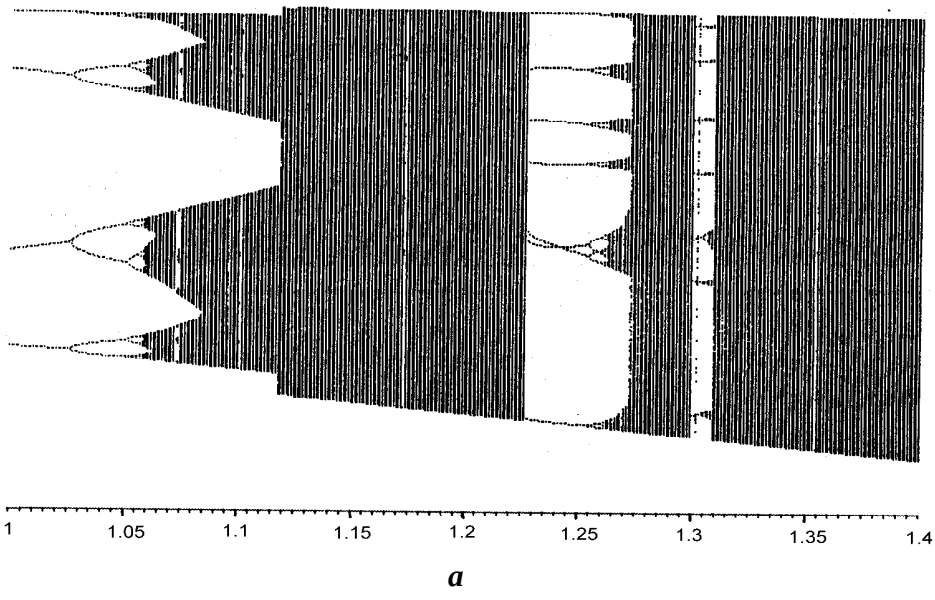


Figure 2.10 Henon map bifurcation diagram vs. control parameter a when b = 0.3.

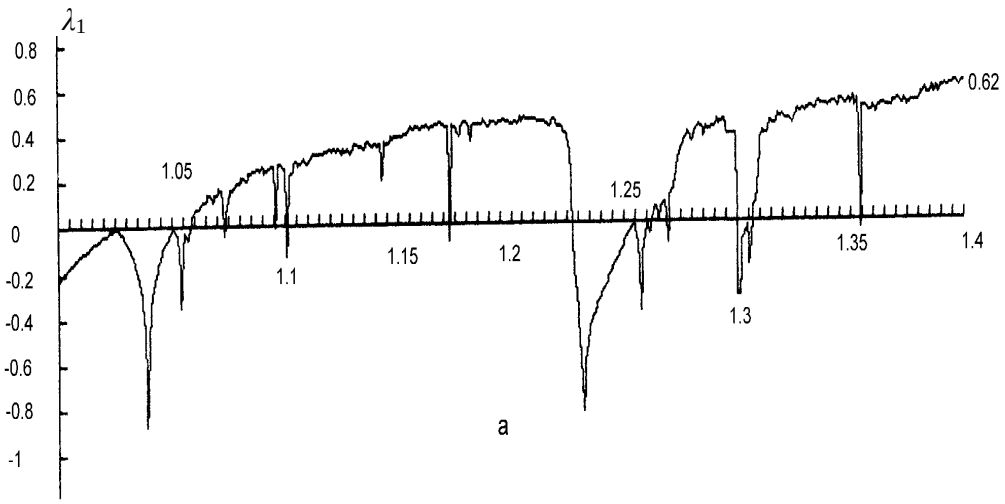


Figure 2.11 Henon map largest Lyapunov exponent vs. *a*, *b* = 0.3

2.9 Conclusion

To summarize, we have reviewed two methods for computing Lyapunov exponents. Each of them has several advantages over the other method. In NRNO method the minimal number of variables is used, rescaling and reorthogonalization are eliminated, and partial Lyapunov spectra can be calculated using fewer equations, there are no difficulties with degenerate Lyapunov spectrum, and global invariance of the Lyapunov spectrum can be explicitly preserved. The method allows a natural split between variables which may be taken advantage of to improve convergence of the exponents. Moreover, the simple form of the final set of equations may prove to be useful in analytic considerations as well. In low dimensional systems the GSR method converges quickly and in order in the exponent spectrum and partial Lyapunov spectrum can be calculated. While with NRNO method our experience shows that exponents mix up and appear out of order.

CHAPTER 3

JOSEPHSON JUNCTION

3.1 Introduction

In this chapter we introduce the Josephson junction at the level required to understand the nonlinear effects. The ac and dc Josephson effects and the resistively and capacitively shunted junction (RCSJ) model for the Josephson junction are introduced. Then, the nonlinear dynamics of the zero-voltage state, including tunneling from the zero-voltage state and microwave activation is discussed. The junction's behavior is demonstrated using 2-dim and 3-dim phase portraits, the bifurcation diagram, the Lyapunov spectrum and a complexity measure. For example, we show how a range of variation of microwave radiation intensity causes system to behave periodically and chaotically. We demonstrate how no matter what initial parameter is fed into the system the output in the periodic region is always the same. In the chaotic region initial conditions matter because they affect the output. The Josephson junction becomes unstable and unpredictable under certain conditions in the presence of microwave radiation. The output signal of the system becomes noise like in the chaotic region. The noise-like signal in some cases may be desired depending on the application of the device.

3.2 The Josephson Effect

Superconductivity occurs when an ensemble of Cooper pairs spontaneously form the synchronized state, known as a Bose-Einstein condensate, and passes through a metal carrying electrical current without resistance. In a pair of superconductors separated by an insulating oxide barrier some 10 or 20 atomic layers thick, in each superconductor, the electron-phonon

interaction leads to the formation of Cooper pairs. In 1962, Josephson realized that these electron pairs can tunnel through the barrier, resulting in an electric current in the absence of any potential difference across the barrier [36]. Because no energy is required to transfer an electron pair from one side of the barrier to the other, the quantum ground states of the junction in which different numbers of pairs have been transferred are degenerate. That is to say, weakly coupled superconductors would exhibit dissipation-less and coherent Cooper pair tunneling. The condensate wave function is coherent not only in either side of the barrier separately but throughout the entire system (including the oxide layer). While the amplitude of the wave function decays inside the oxide, it appears to be sufficient to lock the phases of either side of the oxide barrier. The quantization of magnetic flux is just the requirement that φ be the phase of a single-valued function, so that it may change only by $2\pi n$ on going round a ring. If a voltage is applied across a Josephson junction, taking a pair across will change the energy by -2eV. The variation of the phase in time is given by the Josephson equation:

$$\hbar \frac{\partial(\Delta\varphi)}{\partial t} = 2eV, \tag{3.1}$$

which is just the Einstein equation relating the frequency to the energy of a pair. The tunneling current given by combining this with the dc effect is:

$$J = J_0 \sin\left(\Delta\varphi - \frac{2eVt}{\hbar}\right). \tag{3.2}$$

This is called the ac Josephson effect. That is, an applied dc voltage leads to an alternating current of frequency proportional to the voltage. This ac supercurrent will therefore oscillate with frequency, $f = 2e/h = 483593913\ MHz/V$. This current oscillation is based on fundamental constants and is material independent, and thus makes the Josephson effect a perfect candidate for use as a voltage standard [38] which is developed as natural consequence of extremely

precise measurements of 2*e*/*h* via the Josephson effect, and the recognition that a Josephson junction is a precise frequency-to-voltage converter. It also permits a determination of *e/h* from the relationship between voltage and the frequency of the photon emitted when an electron pair crosses the barrier. In his original derivation of the phase-dependent tunneling supercurrent, Josephson used perturbation theory to treat the transfer term in the Hamiltonian for the tunneling problem but kept track of the phases of the wave functions of either side of the junction. In fact, the phase-dependent supercurrent can be derived as a conserved current from the gauge invariance of the Hamiltonian for the tunneling problem, without invoking perturbation theory.

The Cooper pair current depends on the gauge invariant phase difference of the superconducting wave functions on both sides of the barrier through which the superconductors are coupled, that is $\varphi = \varphi_0 - (2\pi/\Phi_0) \int \mathbf{A}.d\mathbf{L}$. Since a gauge transformation changes the phase but not the amplitude of the pair wave function, it follows that the Josephson voltage-frequency relation $V = (h/2e)f$ is intimately related to flux quantization. That is in (Eq. 3.2) we may write $V=(f/c)\Phi_0$ and $\Delta\varphi = \left(\frac{2e}{\hbar c}\right)\Phi$, where $\Phi_0 = \frac{hc}{2e} = 2.067850654 \times 10^{-15} Wb$, is the magnetic flux quantum. Φ is the total flux due to external fields and due to currents in a closed circuit. Thus the Josephson current varies with flux according to the relation:

$$J = J_0 \sin\left(\frac{2e\Phi}{\hbar c} - \frac{ft}{c}\Phi_0\right). \tag{3.3}$$

This relation provides the basis for one family of Josephson junction devices, called SQUID (superconducting quantum interference device). The SQUID is a double-junction quantum interferometer formed from two Josephson junctions mounted on a superconducting ring.

Magnetic field is applied normal to the plane of the ring. In this device we may write the phases in $\Delta\varphi = \left(\frac{2e}{\hbar c}\right)\Phi$ as

$$\begin{cases} \varphi_1 = \varphi_0 + \left(\frac{e\Phi}{\hbar c}\right) \\ \varphi_2 = \varphi_0 - \left(\frac{e\Phi}{\hbar c}\right) \end{cases},$$

and the total current, which is the sum of J_1 and J_2, as:

$$J_T = J_0\left[\sin\left(\varphi_0 + \frac{e\Phi}{\hbar c}\right) + \sin\left(\varphi_0 - \frac{e\Phi}{\hbar c}\right)\right] = 2J_0 \sin\varphi_0 \cos\left(\frac{e\Phi}{\hbar c}\right). \tag{3.4}$$

The critical current, that is current at which voltage first appears, is therefore a periodic function of the applied flux, with a period equal to one magnetic flux quantum.

3.3 The Josephson Junction Structure

Josephson Junctions are made of weak connections between superconductors through which the Josephson effects are realized. Historically, superconductor insulator-superconductor tunnel junctions have been used to study the Josephson effect, primarily because these are physical situations for which time-averages go to zero except when detailed calculations can be made. However, the Josephson effect is not necessarily a tunneling phenomenon, and the Josephson effect is indeed observed in other types of junctions, such as the superconductor-normal metal-superconductor junction. A particularly useful Josephson junction, the point contact, is formed by bringing a Josephson tunnel sharply pointed superconductor into contact with a blunt superconductor. The critical current of a point contact can be adjusted by changing the pressure of the contact. The low capacitance of the device makes it well suited for high-frequency applications. Thin-film microbridges form another group of Josephson junctions. The simplest microbridge is a short narrow construction (length and width on the order of a few

micrometers or smaller) in a superconducting film known as the Anderson-Dayem bridge. If the microbridge region is also thinner than the rest of the superconducting film, the resulting variable-thickness microbridge has better performance in most device applications. If a narrow strip of superconducting film is over-coated along a few micrometers of its length with a normal metal, superconductivity is weakened beneath the normal metal, and the resulting microbridge is known as a proximity-effect or Notarys-Mercereau microbridge. Among the many other types of Josephson junctions are the superconductor-semiconductor-superconductor and other artificial-barrier tunnel junctions, superconductor-oxide- metal-superconductor junctions, and the so-called SLUG junction, which consists of a drop of lead-tin solder solidified around a niobium wire. HYPRES Inc. has developed and sustains several fabrication processes for superconductor electronics [37]. It currently offers three processes with three different critical current densities of Nb/AlOx/Nb trilayer: 0.3 $\mu A/\mu m^2$, 10 $\mu A/\mu m^2$ and 45 $\mu A/\mu m^2$. The Josephson junctions can be interconnected into circuit configurations using four superconducting layers (junction base electrode (layer M1), two Nb wiring layers (layers M2 and M3) and superconducting Nb ground plane (layer M0). One normal metal layer is used to provide medium-value resistors, which can be used for shunting Josephson junctions, current distribution and other applications. The sheet resistance of this layer is given in the Table 3.1 for all three processes. Silicon dioxide is deposited to provide insulation between the conducting layers. Anodization of the base electrode of the trilayer provides additional insulation to Josephson junctions. Their standard fabrication process uses 6-inch (150 mm) diameter oxidized Si wafers. In the fabrication process, HYPRES uses only refractory materials, with the exception of a Ti/Pd/Au metallization layer used primarily for contact pads. Niobium is used as the superconducting material due to its comparably high critical temperature, electrical and thermal stability, and ability to be thermally

cycled many times without degradation. Niobium/Aluminum-Oxide/Niobium Josephson tunnel junctions are made by depositing an in-situ trilayer across the entire wafer and subsequently defining junction areas by photolithography and etching. This method yields good uniformity and reproducibility of junction parameters. Table 3.2 lists the physical characteristics of the different types of Josephson junctions for niobium-based superconducting integrated circuits. Some different types of Josephson junctions are illustrated in Figures 3.1.

Table 3.1 HYPRES sheet resistance of the layer for all three processes

Process J_c [μA/μm^2]	**Sheet Resistance at 4.5 K° [Ω]**	**Material**	**T_c [K°]**	**Thickness [nm]**
0.3	2.0 ±0.20	Ti/AuPd/Ti	0.0	100±10
10	1.0 ±0.15	Mo	0.9	70±10
45	2.1 ±0.30	Mo	0.9	40±6

Table 3.2 Josephson junction samples

Name	**Type**	**Junction Area**	**C_j**	**L**	**C**	**Method**
L1	Al/AlOx/Al	60 μm^2	1 pF	10 nH	-	Surface mount
Lc1	Al/AlOx/Al	100 μm^2	4 pF	10 nH	10pF	Surface mount
Hypres1	Nb/AlOx/Nb	100 μm^2	4 pF	10 nH	80pF	On-chip
Hypres2	Nb/AlOx/Nb	100 μm^2	4 pF	8 nH	80pF	Two On-chip

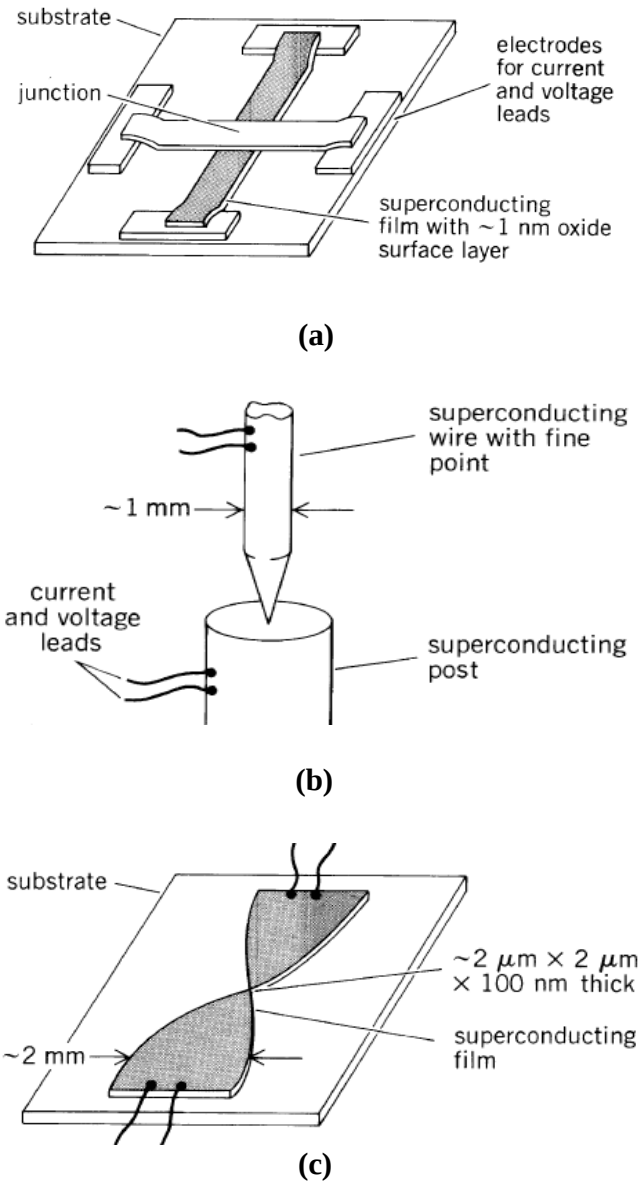


Figure 3.1 Some types of Josephson junctions. (a) Thin-film tunnel junction. (b) Point contact. (c) Thin-film weak link.

The dc current-voltage characteristics of different types of Josephson junctions may differ, but all show a zero-voltage supercurrent, and constant-voltage steps can be induced in the dc characteristics at voltages given by $V=hf/2e$ when an ac voltage is applied. The dc current-voltage characteristics for a weak link (microbridge) and a tunnel junction are compared in Figure 3.2.

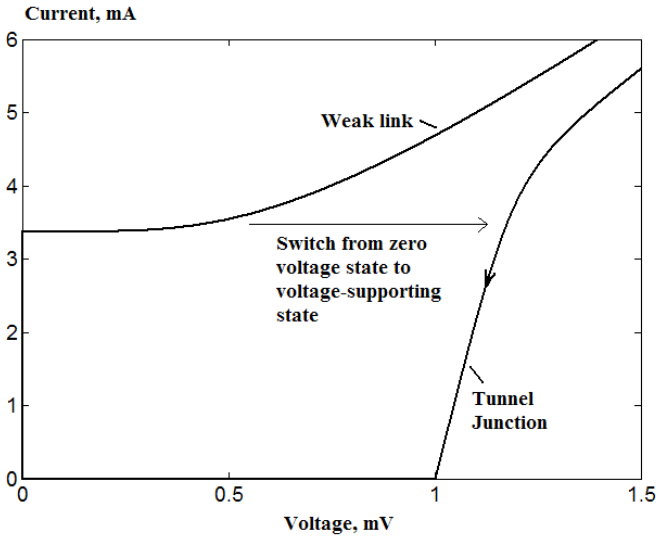


Figure 3.2 DC current-voltage characteristics for a weak and tunnel junction

Josephson junctions, and instruments incorporating Josephson junctions, are used in applications for metrology at microwave frequencies, frequency metrology, magnetometry detection and amplification of electrical signals. Josephson junction devices also have been used in ultrahigh sensitive detectors, and voltage standards. The Josephson junction device has also been applied as a frequency mixer in heterodyne detection and as a generator of high-frequency electromagnetic radiation, as computer elements (with switching time from a zero-voltage to a finite-voltage state of order less than 10^{-9} sec) and in thermometry. Extremely high frequencies (10^{10} – 10^{12} Hz) of the supercurrent provide other important applications, especially in logic circuits and memory cells. Potentially, these high speed and low power digital circuits can be very useful for signal processors and high-performance computers. Also a Josephson junction is used in a qubit realization block for quantum computers. It was for some of these applications and the fundamental nature of the Josephson effect that the discoverer, Brian Josephson, was honored with the 1973 Nobel Prize in Physics.

3.4 Stewart-McCumber Model of a Josephson Junction

When Josephson junction device is driven by a voltage source V_s through a series resistor R, it can have two stable voltage states determined by the electrical parameters, so that $V_j = V_s - iR$ where i is the circuit current, V_j is the junction voltage, and R is the circuit resistance. The function $V_s - iR$ forms the usual load-line which intersects the tunneling (I-V) curve at $V=V_j$ and at $V=0$. For typical currents and tunneling gap voltages, $i \cong 150\mu A$; twice the gap voltage ($2\Delta/e$) is material dependent and ranges from 2.5 mV for Pb and 3.5 mV for Nb to 5.5 mV for NbN electrodes. The significance of having a low conductance voltage region is that when the device is switched from the zero resistance state as in the cryotron to the non-zero voltage state, the drive voltage available to transfer current from the device to a load remains large. This provides a higher switching speed. Thus for properly designed devices, logic delays of about two picoseconds can be achieved. However, once switched to the voltage state the device remains there until its current is dropped to a sufficiently low value. This latching behavior can be exploited in the circuit design. The detailed behavior is best described using the RCSJ model created separately by Stewart and McCumber [39, 40]. The total device currents can be represented as being composed of terms from the quasiparticle currents through an equivalent resistance *R*, an ac displacement current through capacitance *C*, and the Josephson current. In this model, both dissipative quasiparticle tunneling and damping from the wiring are treated as effective shunt impedance which is a function of frequency. For simplicity one can approximate $Z(\omega) = R$.

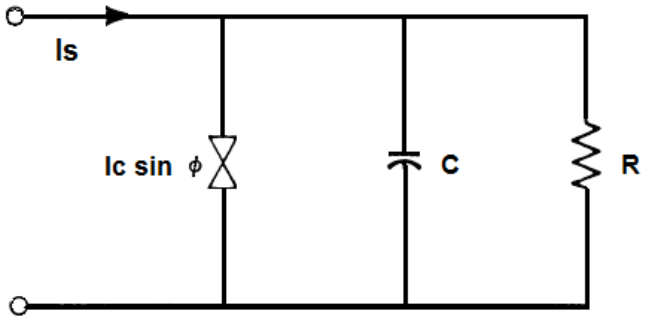


Figure 3.3 Stewart-McCumber RCSJ model of a Josephson junction.

The capacitance between the surfaces of the superconducting electrodes is included in a lumped manner in parallel with the Cooper pair current. The governing equations then become

$$\begin{aligned} V_R &= I_R R, \\ V_C &= \frac{Q}{C}, \\ 2eV_j &= \left(\frac{h}{2\pi}\right)\frac{d\varphi}{dt}, \\ V_R &= V_C = V_j = V. \end{aligned} \tag{3.5}$$

Where $I_j = I_c \sin\varphi$ and I_c is the critical supercurrent characterizing the coupling strength between the two superconducting electrodes. Later when we study tetrode microbridge dynamics, all of the junctions are assumed to be aluminum oxide as a barrier to provide the weak coupling. For this case, the coupling exponentially decreases with increasing barrier thickness. The circuit equation can be written as

$$I_S = I_c + I_R + I_j = C\frac{dV}{dt} + \frac{V}{R} + I_C \sin\varphi. \tag{3.6}$$

Then

$$\left(\frac{h}{2e}\right)\left(\frac{1}{2\pi}\right)C\frac{d^2\varphi}{dt^2} + \left(\frac{h}{2e}\right)\left(\frac{1}{2\pi R}\right)\frac{d\varphi}{dt} + I_C \sin\varphi = I_S,$$

$$(\frac{\Phi_0}{2\pi})C\frac{d^2\varphi}{dt^2}+(\frac{\Phi_0}{2\pi})(\frac{1}{R})(\frac{d\varphi}{dt})+I_C\sin\varphi=I_S,$$

(3.7)

With the Lagrangian and Hamiltonian given by

$$L=\frac{1}{2}\left(\frac{\Phi_0}{2\pi}\right)^2 C\left(\frac{d\varphi}{dt}\right)^2+(\frac{\Phi_0}{2\pi})I_C\cos\varphi-\left(\frac{\Phi_0}{2\pi}\right)\varphi I_S,$$

$$H=\left(\frac{d\varphi}{dt}\right)p-L=\left(\frac{p^2}{2m}\right)-m\omega_J^2\cos\varphi-\left(\frac{\Phi_0}{2\pi}\right)\varphi I_S,$$

$$p=\frac{\partial L}{\partial(\dot{\varphi})}-L=\left(\frac{\Phi_0}{2\pi}\right)^2 C\left(\frac{d\varphi}{dt}\right),$$

$$m=\left(\frac{\Phi_0}{2\pi}\right)^2 C,\quad \omega_J=\sqrt{\frac{2\pi I_C}{\Phi_0 C}}.$$

(3.8)

ω_J is the zero-bias small oscillation frequency or Josephson plasma frequency, which is a natural mode of oscillation of the system. These equations describe the motion of a phase particle of mass m in the tilted washboard potential. This result presupposes a junction small enough to allow I_c to be uniform. The magnetic self-field and external fields combined are smaller than a flux quantum Φ_0. The washboard potential model of a single Josephson junction is shown in Figure 3.5. The distance between adjacent wells is 2π. The bias current, I_s, controls the tilt of the washboard and thus the barrier height ΔE. The effect of a small R is to damp the motion of the phase particle. The tilted washboard potential has two types of classical states, metastable states with $d\varphi/dt=0$ called the zero-voltage state and unbound voltage states with continuously increasing. The bias current I_s controls the energy barrier ΔE between the zero-voltage and voltage state, as well as the tilt of the washboard potential. Once the bias current exceeds the critical current there are no longer any bound states subject to viscous damping of coefficient $1/RC$ with $\tau = RC$ as the damping time.

$$U(\varphi) = \frac{2\pi}{\Phi_0}\left(\frac{1}{C}\left(I_s - I_c \sin\varphi\right)\right),$$

$$\frac{d^2\varphi}{dt^2} + \frac{1}{RC}\frac{d\varphi}{dt} = U(\varphi), \tag{3.9}$$

For $I_s < I_c$, there is a static solution with $\varphi < \frac{\pi}{2}$, wherein $\frac{d\varphi}{dt} = 0$ and $\frac{d^2\varphi}{dt^2} = 0$, therefore $U(\varphi) = 0$. As I_s is increased so that $|I_s| > |I_c|$ then φ increases with time and $V = \frac{\Phi_0}{2\pi}\left(\frac{d\varphi}{dt}\right) \neq 0$.

In the steady state, $\frac{d^2\varphi}{dt^2} = 0$ and thus $I_j = I_c \sin\varphi = I_c \sin(2\pi/\Phi_0 \int V\mathrm{d}t)$, for constant V is:

$$I_j = I_c \sin\varphi = I_c \sin\left(\frac{2\pi Vt}{\Phi_0}\right) = I_c \sin\left(\frac{2eVt}{h}\right) \tag{3.10}$$

For a device with a relatively large *RC* product i.e. small damping, it is apparent that once it is in the voltage state, reducing I_s such that $I_s < I_c$ does not return the junction to the zero resistance state. Referring to the washboard potential model, one sees that the device dynamics are indeed complex. Starting from an initial drive current $I_s = 0$, the particle stays trapped in a potential well, $V = 0$, since $d\varphi/dt = 0$. As I_s increases, then φ increases until finally the particle falls out of the well and proceeds to the subsequent ones, accelerating until damping losses stabilize the average speed. If one now reduces I_s to less than I_c, $d\varphi/dt$ does not go to zero that is V ≠ 0 until a sufficiently low I_s is reached whose value is determined by the viscous damping term. Thus the device is hysteretic, which is both an advantage and a problem. The fact that it latches provides for a stable two-state system, but it also removes control of the state from the input signal. As one changes the RC product, most commonly by changing R, the *I-V* curve can be

made to transition to a non-hysteretic form which in some applications such as magnetometers is desirable.

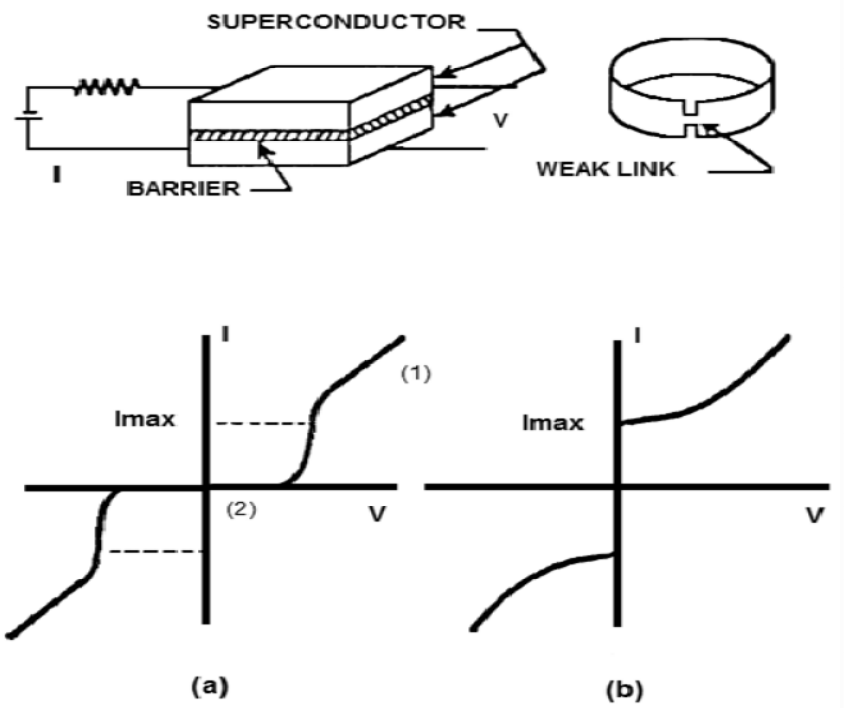


Figure 3.4 Josephson Junction I-V characteristic curves (a) hysteretic (b) non-hysteretic.

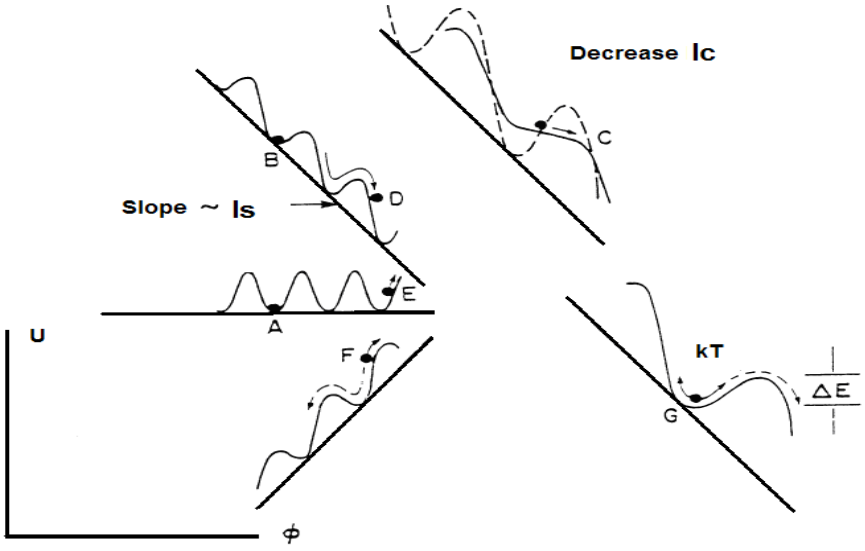


Figure 3.5 Washboard potential model of a single Josephson junction. The distance between adjacent wells is 2π. The bias current controls the tilt of the washboard and thus the barrier height ΔE. The effect of a small R is to damp the motion of the phase particle.

3.5 Chaos in a Josephson Junction

A Josephson junction device has nonlinear dynamical characteristics and displays a wide variety of nonlinear phenomena. The dynamics of tunnel junctions can be described by RCSJ model. A Josephson junction circuit with an inductor is an autonomous system that has a model circuit consisting of RLC components with nonlinearity. This is equivalent to a Duffing oscillator which can exhibit chaotic oscillations. Chaotic oscillations are also observed when an external microwave oscillation is applied to a Josephson junction at the characteristic frequency of the Josephson junction [41]. In addition chaotic oscillations have been observed in mutually coupled two Josephson-junction devices through capacitors, resistors, and inductors [42]. In non-autonomous Josephson systems, chaotic behavior can be due to the mixing of the external and internal frequencies. Other devices with many Josephson junctions and inductors like a SQUID, exhibit various nonlinear dynamics [43]–[50]. There have been some studies on the use of different coupling configurations in order to obtain chaotic oscillations, such as one-dimensional linear coupling and two-dimensional coupling configurations [51]–[53]. To generate ultrafast chaotic oscillations for the purpose of engineering applications of random signal generation, Josephson junction devices are required to be small, simple, no capacitance, and autonomous systems. But there have been no reports of Josephson junction devices that satisfy all the above-mentioned conditions. However, in chapter five, we investigate the dynamics of a new class of systems that are made of few Josephson microbridges that satisfy the aforementioned conditions. But first we study the behavior of a single Josephson junction under microwave radiation. This two-terminal device exhibits extremely rich dynamics and displays a wide variety of interesting nonlinear phenomena. The next set of simulations establishes the importance of control parameter space values on the dynamic states of the single junction system. Consider an external

signal is applied in the form of microwave radiation with amplitude I_{RF} and frequency ω_d, that is, $I_S(t) = I_{RF}\cos(\omega_d t)$. Replacing m and ω_J, in a single Josephson junction dynamic equation we write:

$$\begin{aligned}
&(\frac{\Phi_0}{2\pi})C\frac{d^2\varphi}{dt^2}+(\frac{\Phi_0}{2\pi})(\frac{1}{R})(\frac{d\varphi}{dt})+I_C\sin(\varphi)=I_S,\\
&\frac{d^2\varphi}{dt^2}+(\frac{1}{RC})(\frac{d\varphi}{dt})+\omega_J^2\sin(\varphi)=(\frac{2\pi}{\Phi_0})(\frac{I_{RF}}{C})\cos(\omega_d t),\\
&\omega_J=\sqrt{2\pi I_C/\Phi_0 C}=\sqrt{2eI_c/\hbar C}.
\end{aligned} \tag{3.11}$$

The nonlinear solutions of equation 3.3 are expressed in terms of a new time scale $\theta = t/\alpha$ and frequency scale $\omega_0 = \alpha\omega_J$. Thus by transforming operators are

$$\begin{aligned}
\frac{d^2\varphi}{dt^2}&=\left(\frac{1}{\alpha^2}\right)\frac{d^2\varphi}{d\theta^2}\\
\frac{d\varphi}{dt}&=\left(\frac{1}{\alpha}\right)\frac{d\varphi}{d\theta}
\end{aligned} \tag{3.12}$$

Then the system equation become

$$\frac{d^2\varphi}{d\theta^2}+k\frac{d\varphi}{d\theta}+\omega_0^2\sin(\varphi)=g\cos(\alpha\omega_d\theta), \tag{3.13}$$

where $\tau = RC$, $k = \alpha/\tau$ and $g = (2\pi\,\alpha^2 I_{RF}/\,\Phi_0 C)$. Substituting $z = \omega_d\theta$ we have:

$$\begin{cases}
\dfrac{d\varphi}{d\theta}=y\\
\dfrac{dy}{d\theta}=-ky-\sin(\varphi)+g\cos(z),\\
\dfrac{dz}{d\theta}=\omega_d
\end{cases} \tag{3.14}$$

Recall that φ, the phase difference between quantum states; k, the dissipation constant in the system; ω_d, the normalized microwave frequency; g, the normalized amplitude of the microwave frequency and $V = d\varphi/dt$, the voltage. Equations are integrated numerically with initial condition $V(0) = d\varphi/d\theta\big|_{\theta=0} = 0$. A typical electrical characteristics of junction are set to R = 4 Ω, C = 5pF, I_c = 100 μA. We choose $\alpha = 10^{-11}$ such that α/τ = 0.5 and ω_0 = 2.53. The integration time step was ε = 0.1 and 1.5×10^4 iterations were performed after discarding first 2×10^3 transient solutions. Typical phase space portrait shows some of the solutions for different values of driving amplitude, g when dissipation coefficient k = 0.5, and driving flux frequency ω_d = 2/3 are kept fixed. These are illustrated in Figure 3.6. Obviously, the Josephson junction system demonstrates periodic and chaotic behavior for certain values of radiation amplitude.

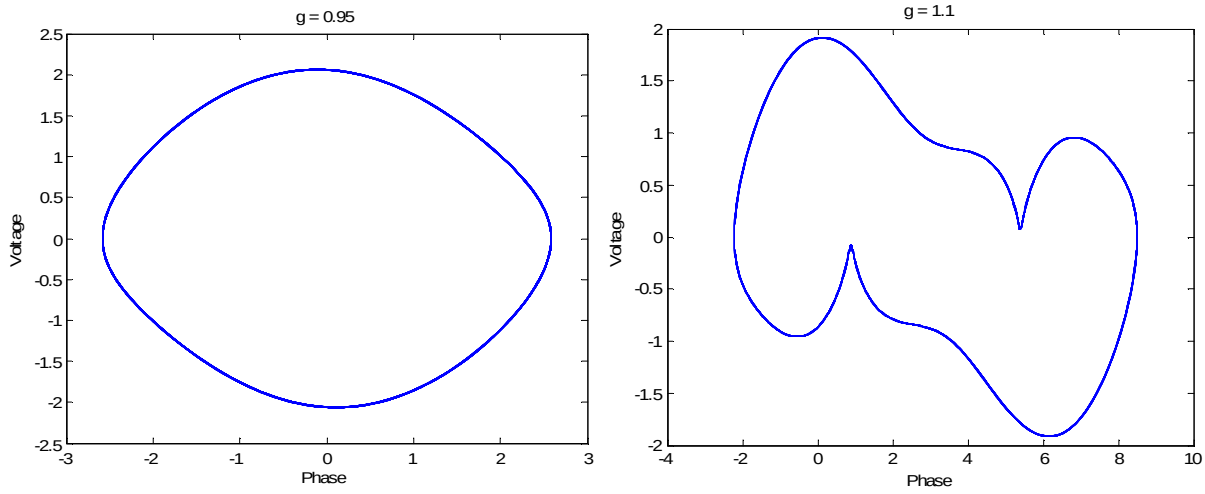


Figure 3.6 Josephson junction system phase portraits for different applied driving amplitudes (g). Dissipation coefficient, k = 0.5; driving frequency, ω_d= 2/3; 1.5×10^4 iterations; 2×10^3 transient states.

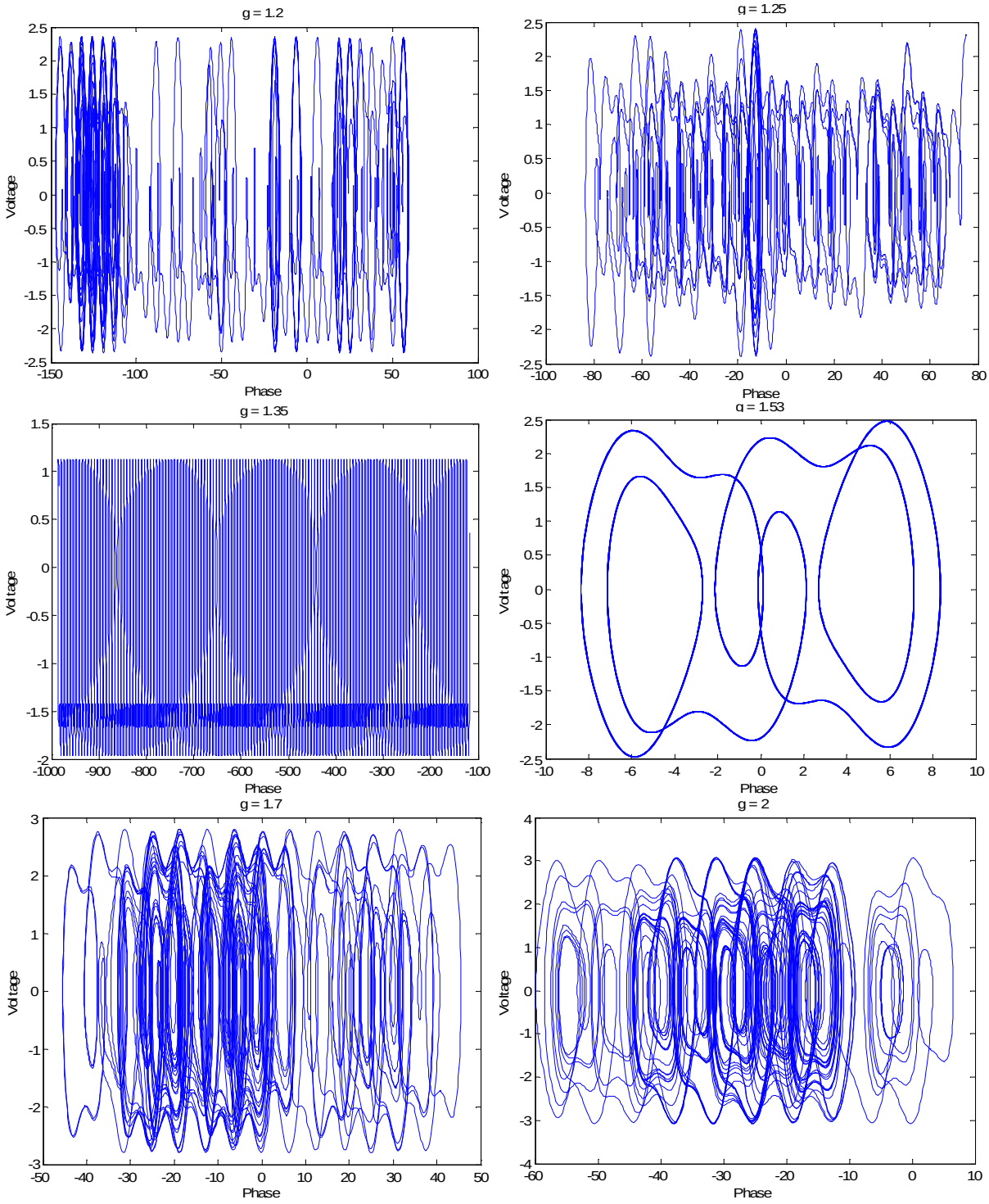


Figure 3.6, cont. Josephson junction system phase portraits for different applied driving amplitudes (g). Dissipation coefficient, $k = 0.5$; driving frequency, $\omega_d = 2/3$; 1.5×10^4 iterations; 2×10^3 transient states.

Bifurcation diagram shows the possible long-term values a variable of a system can obtain as a function of a parameter of the system, which in our case is the amplitude of the microwave radiation. For the bifurcation diagram in Figure 3.9 the amplitude of the microwave radiation (g) is gradually increased in the range $0.9 \leq g \leq 1.5$ with increment by $\Delta g = 0.005$ to the next value. For each parameter value, 5×10^4 iterations are used and the first 10^3 responses are discarded. From this graph we can see that the system's output is completely different for every calculation. Chaotic states are characterized by strange attractors in phase space displaying an infinite lattice structure with the spatial periodicity of the potential. Although the system is in its chaotic region when observed closely the system has windows of periodic behavior. The bifurcation diagram further confirms our previously obtained results for the periodic and chaotic regions. We were able to observe how the systems behavior bifurcates from one cycle to two cycle and four cycle behavior to chaotic behavior. From this diagram we can observe that when g = 1.0 the system is in its periodic behavior, while for g = 1.2 the system is in its chaotic regime. When g = 1.48 the system is periodic with four cycles. We have calculated the Lyapunov exponents by GSR, introduced in chapter two. In the first trial microwave radiation amplitude is used as a control parameter to vary the dynamical state of the Josephson junction, but the microwave radiation frequency ω_d and dissipation coefficient are kept fixed at 2/3 and 0.5 respectively. Integration is performed using the fourth-order Runge–Kutta algorithm, with a fixed step size of 0.01, which resulted in 4×10^5 iterations. The result is shown in Figure 3.10 (a). In this case the Lyapunov spectrum is $(\lambda_1, \lambda_2, \lambda_3)$. Stable periodic attractors have only zero and negative Lyapunov exponents. For example consider $(\lambda_1 = 0, \lambda_2 = -, \lambda_3 = -)$ where the zero corresponds to the limit-cycle trajectory itself or $(\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = -)$ for an attracting 2-torus. Chaotic attractors have just one finite positive Lyapunov exponent. For this system there is only

one finite positive Lyapunov exponent. The symbolic Lyapunov spectrum of the form $(\lambda_1 = +, \lambda_2 = 0, \lambda_3 = -)$, the zero corresponds to the chaotic trajectory itself, positive for expanding and negative for contracting trajectories. The precision of the calculation of the Lyapunov exponent(s) depends on many factors in numerical simulation such as the number of integration steps, the value of time-step, and the number of the reorthonormalization steps (GSR steps). This sensitivity of the Lyapunov exponent computation on the above factors increases for dynamic systems with higher dimensions. In Josephson junction systems, the addition theorem is used to determine the optimum value of the GSR steps for fixed integration steps of 2×10^5 on a chaotic attractor with driving amplitude and flux frequency of g =3.8, ω_d = 0.5 and dissipation coefficient k = 0.5. This simulation is shown in Figure 3.7. Obviously, the best optimum choices for the GSR step are between 4 to 7 intermediate integration steps. In addition, the convergence of Lyapunov exponent in terms of the simulation integration numbers is examined. This is done by plotting the largest Lyapunov exponent versus the iteration numbers starting from 2×10^5 and extending to 7.2×10^6. The GSR step is set to five for each iterations number. It is observed from the results in Figure 3.8 that the most cost effective choice is the 4×10^5 trial with the minimum error in LE estimation.

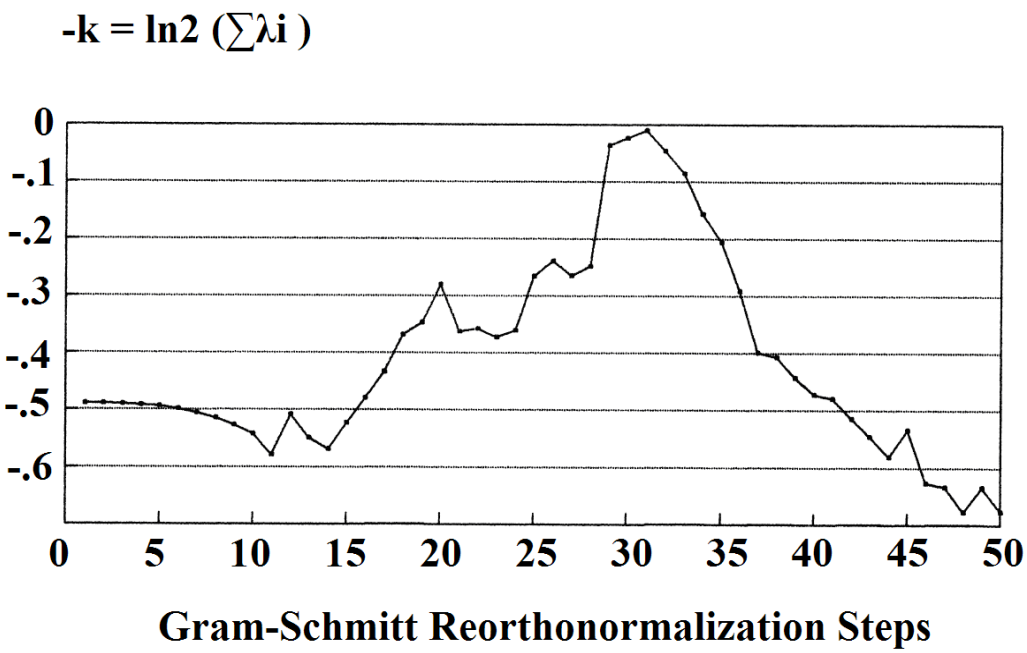


Figure 3.7 Josephson junction system LE sum (ln2) $\sum\lambda_i$ plotted vs. GSR steps. Control parameters: g = 3.8, k = 0.5, ω_d = 2/3 over 2×10^5 iterations.

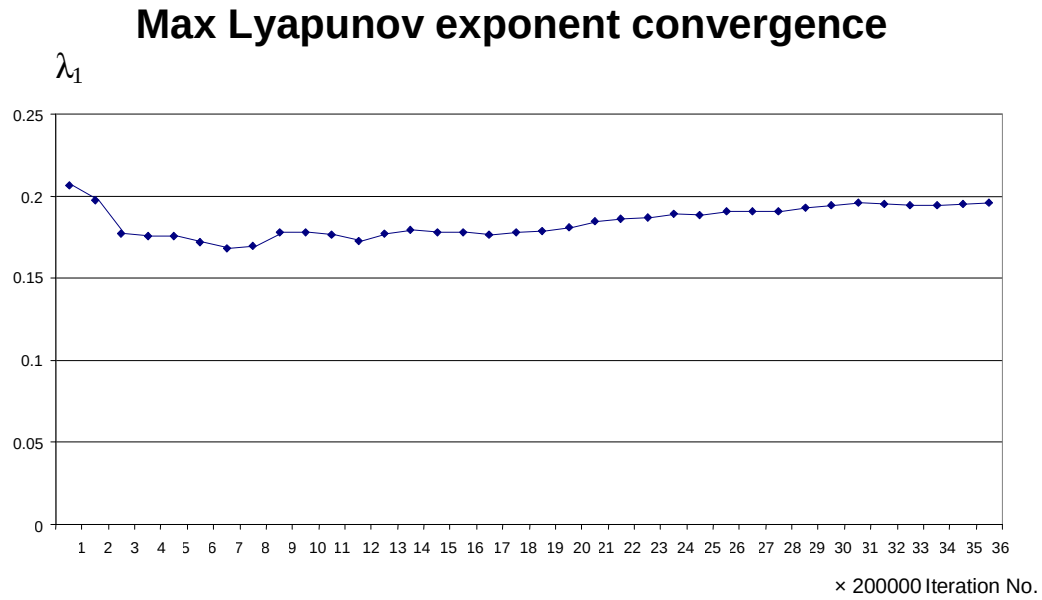


Figure 3.8 Josephson junction system maximum LE vs. number of iterations. Control parameters: g = 3.8, k = 0.5, ω_d = 2/3 and GSR steps = 5.

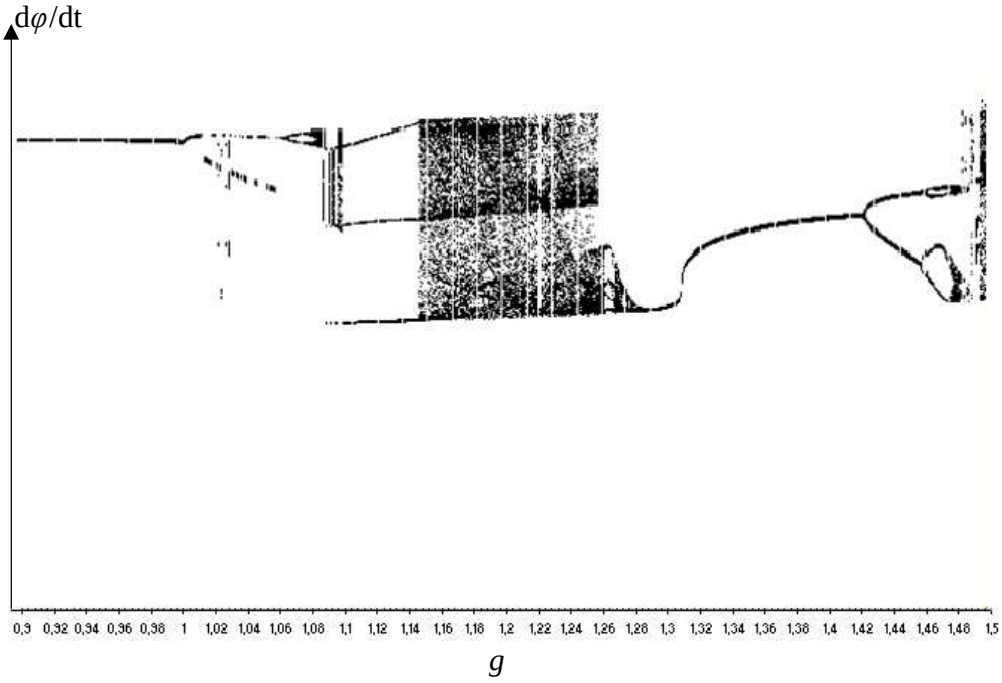
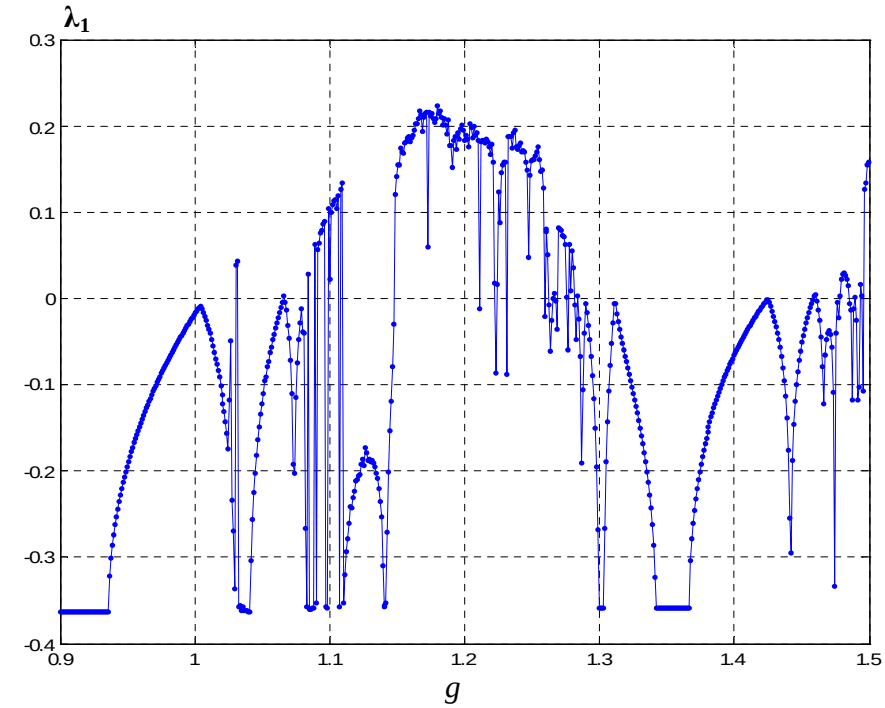
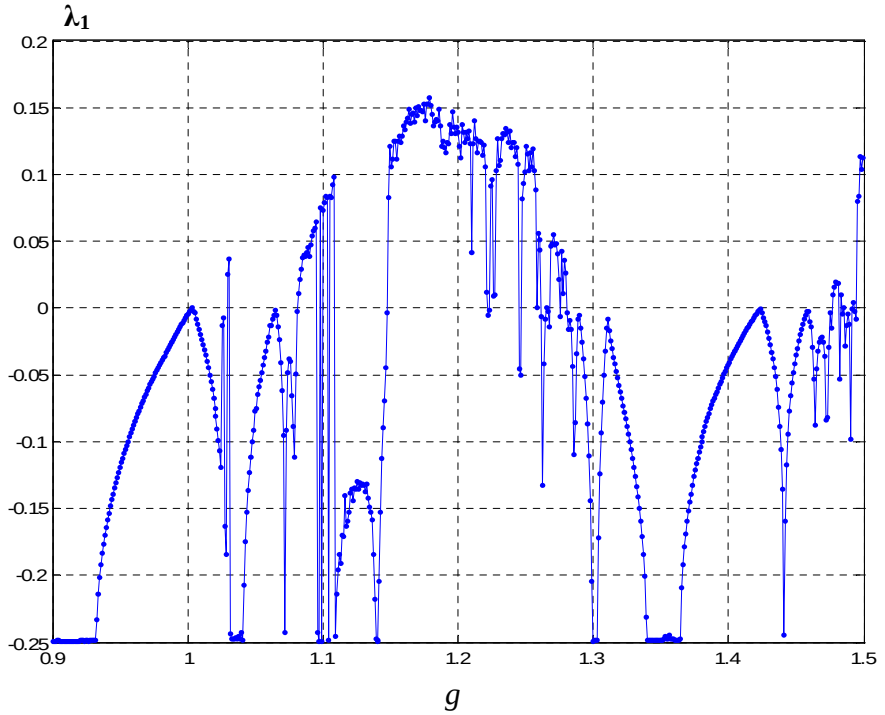


Figure 3.9 Josephson junction bifurcation diagram for microwave radiation amplitude values $0.9 \leq g \leq 1.5$ [$\Delta g = 0.001$], k=0.5, $\omega_d = 2/3$.



(a)



(b)

Figure 3.10 Josephson junction system largest Lyapunov exponents calculated by (a) The GSR method (b) NRNO method. Control parameters: ($0.9 \leq g \leq 1.5$) [$\Delta g = 0.001$], $k = 0.5$, $\omega_d = 2/3$.

Here we provide the formalism for Lyapunov exponent spectrum for a single Josephson junction system by NRNO method. Theoretically for this two-dimensional system we have

$$\frac{d^2\varphi}{dt^2} + k\frac{d\varphi}{dt} + \sin(\varphi) = g\cos(\omega_d t),$$

$$\vec{F}(\vec{X}, \vec{X}_0, \vec{C}) = \frac{d\vec{X}}{dt} = \begin{pmatrix} y \\ -ky - \sin\varphi + g\cos(\omega_d t) \end{pmatrix},$$

$$J = \begin{pmatrix} 0 & 1 \\ -\cos\varphi & -k \end{pmatrix},$$

$$\nabla.\vec{F} = \sum_{i=1}^{2} \lambda_i = -k.$$

The angle matrix, Ω is given by $\Omega = \Psi^{(1,2)} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}$ and $H = \begin{pmatrix} e^{\lambda_1} & h_{12} \\ 0 & e^{\lambda_2} \end{pmatrix}$

The desired equations for λ_1, λ_2 and α are as follows

$$\frac{d\lambda_1}{dt} = \Delta_{11}, \frac{d\lambda_2}{dt} = \Delta_{22}, \frac{d\alpha}{dt} = \omega_1(\dot{\alpha}) = \Delta_{21}.$$

$$\Delta = \tilde{\Omega} J \Omega = \tilde{\Psi}^{(1,2)} J \Psi^{(1,2)} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix},$$

$$\Delta_{11} = \frac{d\lambda_1}{dt} = J_{11}\cos^2\alpha - \sin\alpha\cos\alpha(J_{12} + J_{21}) + J_{22}\sin^2\alpha,$$

$$\Delta_{22} = \frac{d\lambda_2}{dt} = J_{11}\sin^2\alpha + \sin\alpha\cos\alpha(J_{12} + J_{21}) + J_{22}\cos^2\alpha,$$

$$\Delta_{21} = \frac{d\alpha}{dt} = J_{11}\sin\alpha\cos\alpha - J_{12}\sin^2\alpha + J_{21}\cos^2\alpha - J_{22}\sin\alpha\cos\alpha.$$

The parameter values are set to $0.9 \le g \le 1.5$, ω_d = 2/3 and k = 0.5. The above differential equations are numerically integrated forward in time until the desired convergence for the exponents, λ_1/t and λ_2/t is achieved. Evolution equations governing the system include two equations for two Lyapunov exponents and one for the rotation angle. Furthermore, error and convergence analysis was carried out by applying the two methods to the trajectory sampled over time intervals $t_s \ge \varepsilon$ with integration time step $\varepsilon = 0.01$. As a function of t_s, both GSR and

NRNO methods were convergent. GSR method possesses a smaller coefficient for the positive Lyapunov exponent and a larger one for the negative exponent. For this 3-Dim system, execution times for both methods were similar. For high dimensional chaotic systems NRNO method is expected to be more efficient. Figure 3.10 (a) and (b) illustrate the numerical results obtained from the GSR and NRNO methods for the first LE. Figure 3.11 and 3.12 illustrate the complete LE spectrum. Results obtained by NRNO are in complete agreement with values obtained using the GSR approach. Also Figure 3.13 shows the effect of small number of iterations (10^4) on the precision of computation results, compared to Figure 3.13 when the number of iterations is adequate (5×10^5).

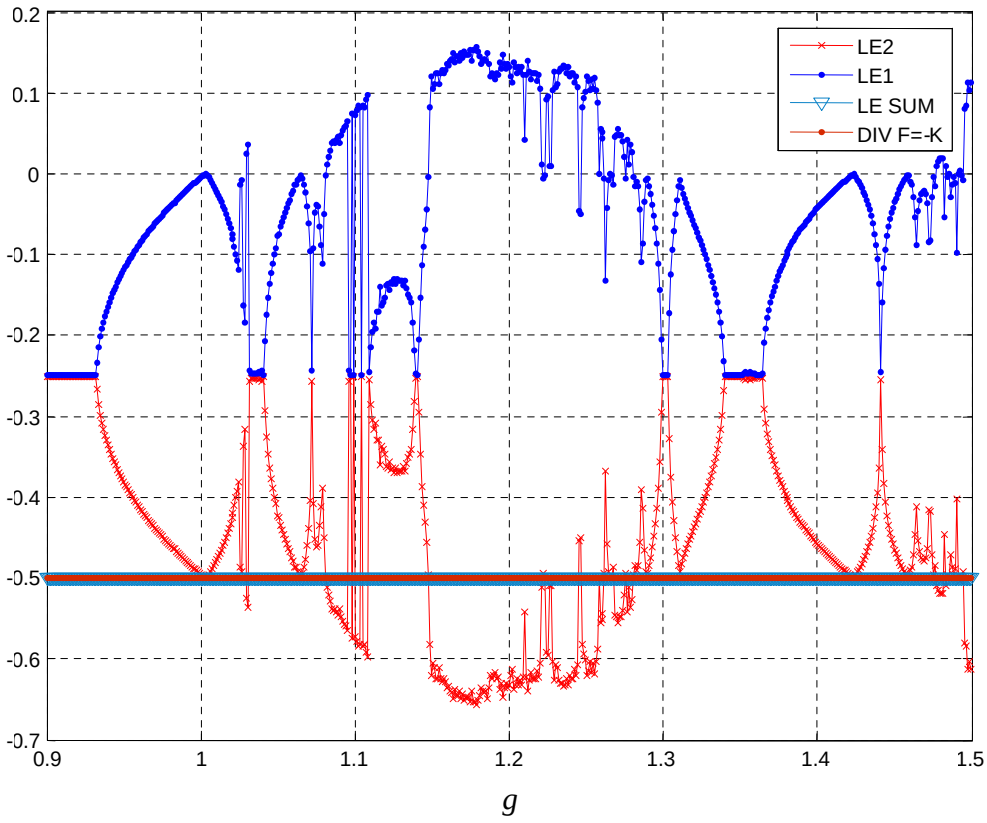


Figure 3.11 Josephson junction system LE spectrum (λ_1, λ_2, λ_3 and $\sum\lambda_i$) by GSR method. Control parameters: $0.9 \leq g \leq 1.5$ [$\Delta g = 0.001$], $k = 0.5$, $\omega_d = 2/3$ over 10^5 iterations after 2×10^4 transients.

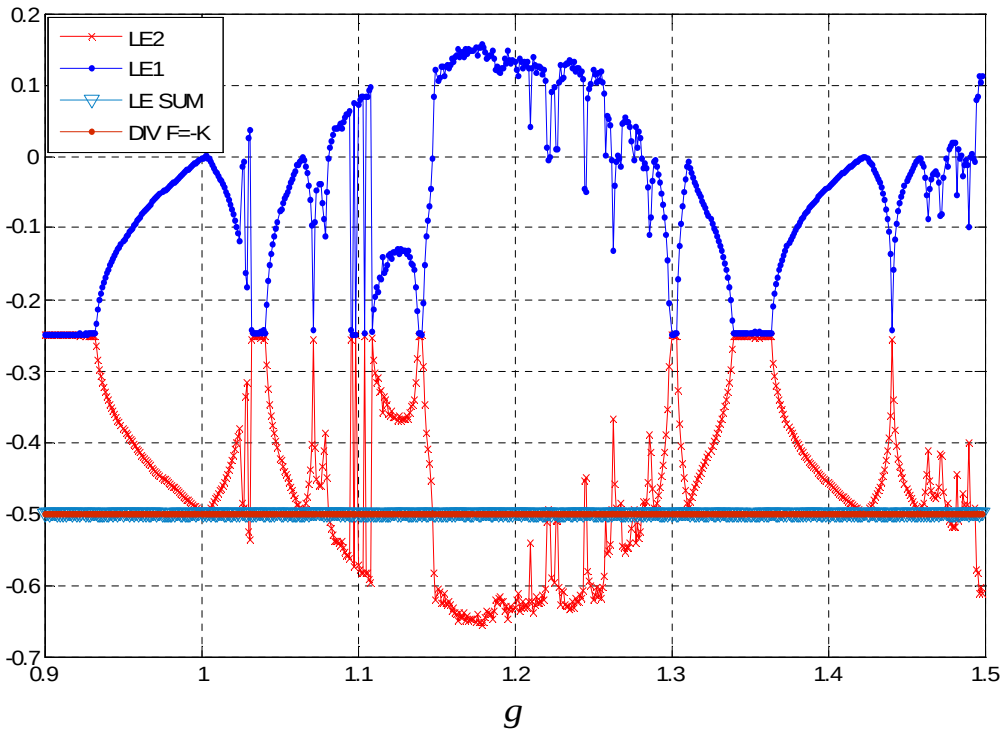


Figure 3.12 Josephson junction system LE spectrum (λ_1, λ_2, $\sum\lambda_i$, and $\nabla.F$) by NRNO method. Control parameters: $0.9 \leq g \leq 1.5$ [$\Delta g = 0.001$], $k = 0.5$, $\omega_d = 2/3$ over 10^5 iterations after 2×10^4 transients.

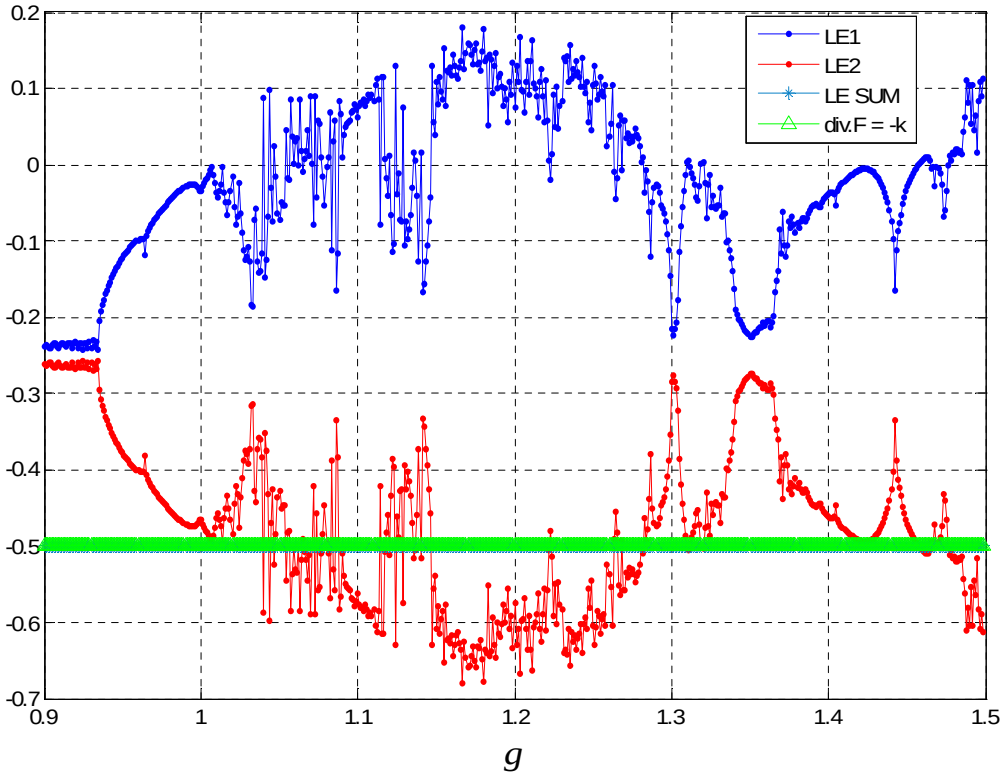


Figure 3.13 Josephson junction system LE spectrum (λ_1, λ_2, $\sum\lambda_i$, and $\nabla.F$) by NRNO method. Control parameters: $0.9 \leq g \leq 1.5$ [$\Delta g = 0.001$], $k = 0.5$, $\omega_d = 2/3$ over 10^5 iterations after 2×10^4 transients.

Next, the study of Josephson junction system LE spectrum focuses on the region of dissipative coefficient $0 \leq k \leq 0.8$ with external driving amplitude $g = 3.8$ and drive frequency $\omega_d = 0.5$ over 5×10^5 iterations for each selected k value. Here we found an interesting existence of chaotic states and periodic states. The chaotic states can be reached by increasing the dissipation coefficient from zero. Figure 3.13 is a composite plot of first, second and third Lyapunov exponents λ_1, λ_2, λ_3 and the div$\boldsymbol{F}$ = ln2 $\sum\lambda_i$ which is checked upon and compared to k for accuracy of Lyapunov exponents computation. In this regime of operation, system ultimately will reach the periodic states as k increases. This means, by increasing k, distinct dynamical modes of chaotic behavior degrade in this region. Samples of the phase portraits of Period-1

states (electrical voltage versus phase) are shown in Figure 3.14 for values of k at 0.25, 0.4, 0.5, 0.6, 0.8, 3.5 and 5. Portraits of chaotic states are shown for values of k equal to 0.05, 0.2, 0.3, 0.7, and 2.9. The external driving amplitude and frequency are set to g = 3.8, ω_d = 0.5 in this trial.

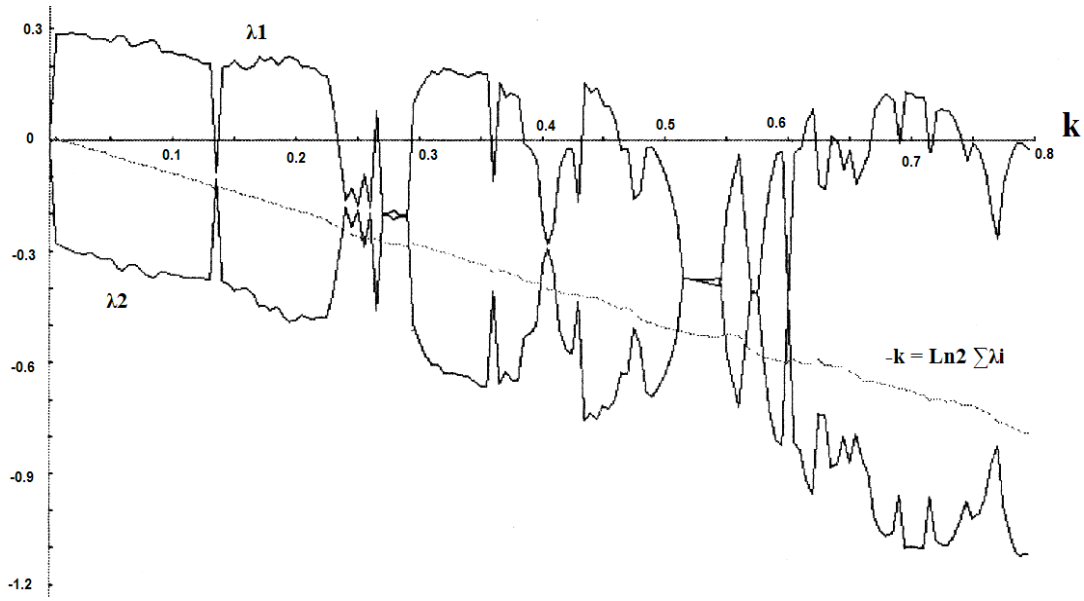


Figure 3.14 Josephson junction system Lyapunov spectrum (λ_1, λ_2, λ_3) and (ln2)$\sum\lambda_i$) vs. Dissipation coefficient $0 \leq k \leq 0.8$, [Δk = 0.01]; microwave amplitude and frequency are g = 3.8 and ω_d = 0.5 over 5×10^4 iterations.

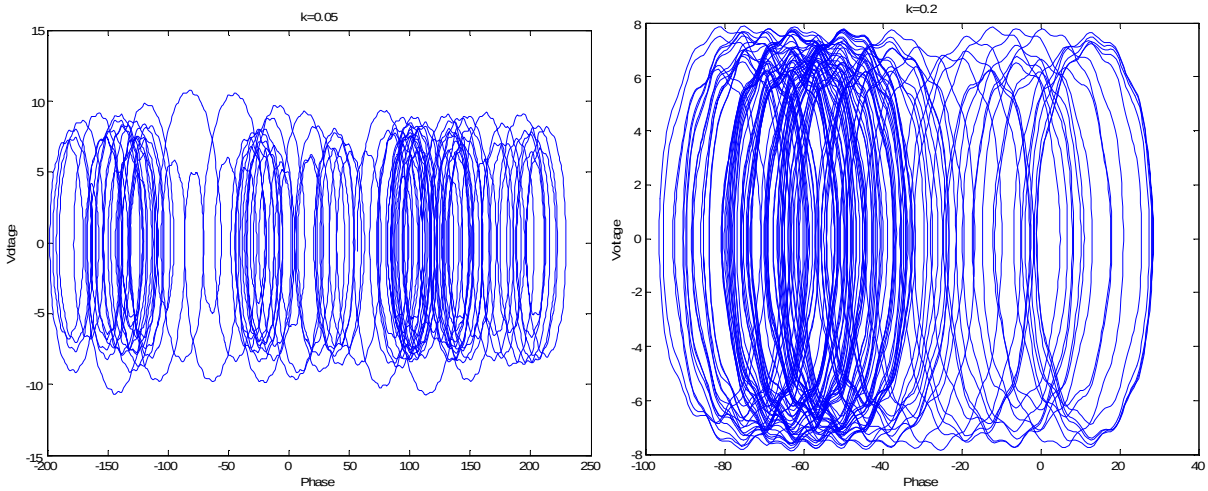


Figure 3.15 Portrait of Josephson electrical voltage vs. phase. Chaotic oscillations correspond to k = 0.05, 0.2. Microwave amplitude and corresponding frequency are g = 3.8 and ω_d = 0.5.

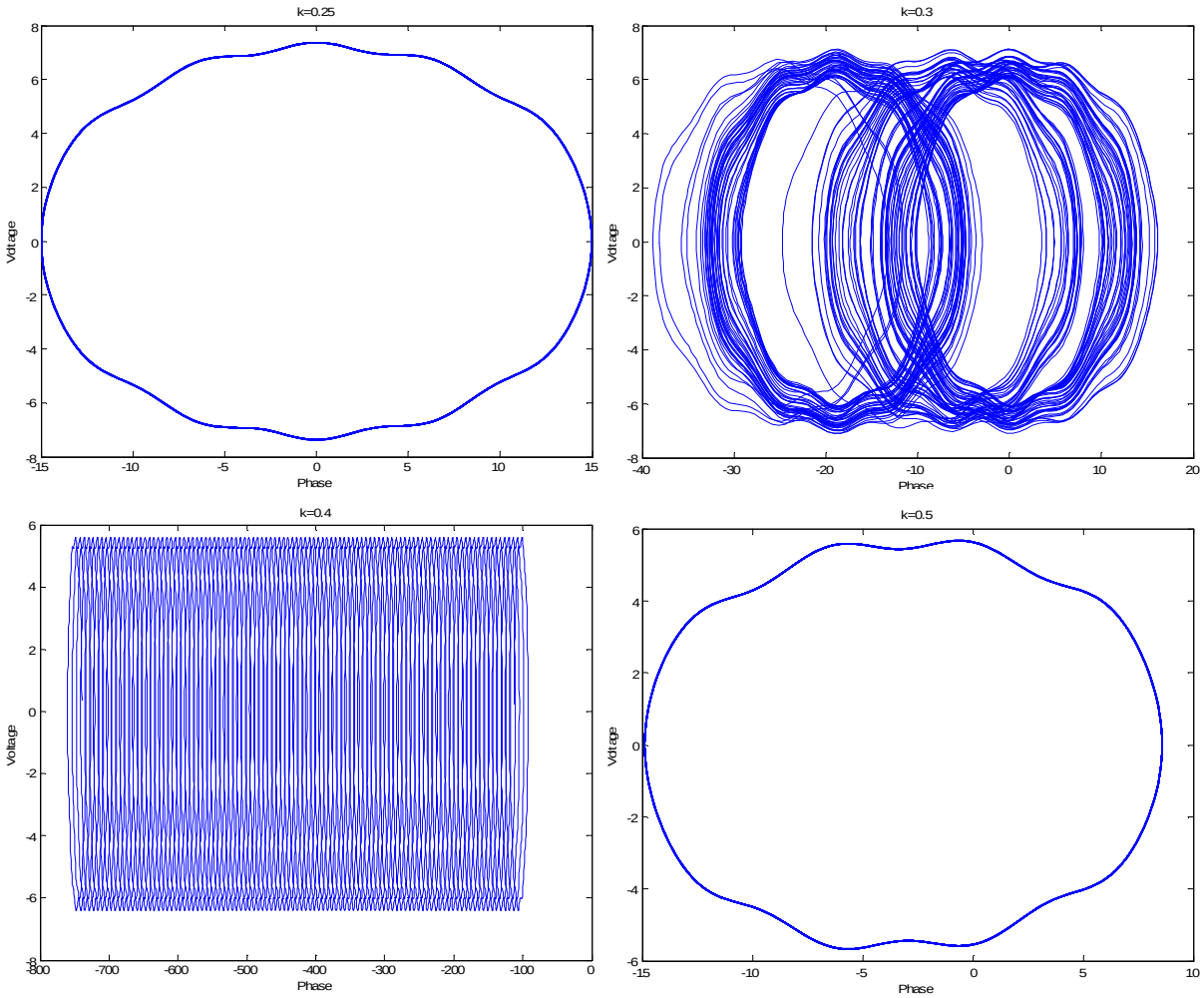


Figure 3.15, cont. Portrait of Josephson electrical voltage vs. phase. Periodic oscillations at $k = 0.25, 0.4, 0.5$ and chaotic oscillation correspond to $k = 0.3$.

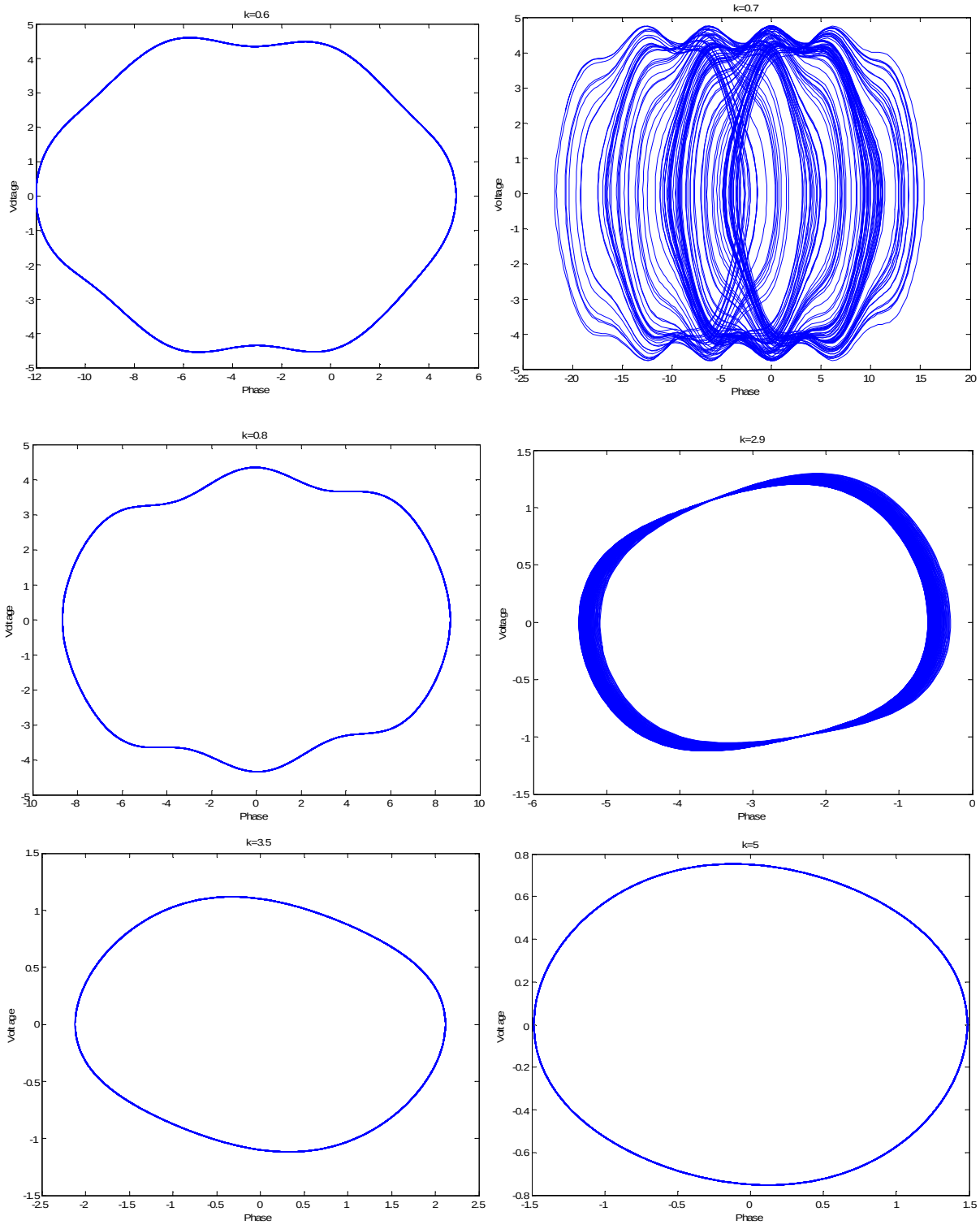


Figure 3.15, cont. Portrait of Josephson electrical voltage vs. phase. Periodic oscillations at k = 0.6, 0.8, 3.5 and 5. Chaotic oscillations correspond to k = 0.7 and 2.9.

To investigate the dependency of the system on the parameters g, k, and ω_d we draw the parametric phase transition diagram. In each case, we use two of the control parameters in a grid with the initial conditions $x = 0.1$, $y = 0.1$. The phase diagram is drawn in the range from 0.9 to 1.5 for the parameter g with $\Delta g = 0.01$, and from 0 to 0.9 for k, with $\Delta k = 0.01$. Thus, the region of phase is subdivided into 90 by 60 cells. Our phase diagram is quantized in step of 0.01 for the g and k axes. The microwave frequency is $\omega_d = 2/3$. Each trial is performed over the 10^5 iterations. 10^4 transients are discarded with integration time step $\varepsilon = 0.01$. We construct the phase diagram of the parameter space by showing the positive values of the first Lyapunov exponent. By exploring the positive second Lyapunov exponent, regions of chaotic responses are identified. The system shows very rich dynamic phases for various values of the parameters g and k. There are regions of periodic states and a periodic doubling state in wide chaotic states. A noteworthy periodic regime, which occurs at fairly high values of k, is the one in which junction phase, φ, undergoes successive 2π rotations in phase with the driving frequency, corresponding to the periodic motion of the particle from one potential energy well to another. This kind of behavior represents the running periodic solutions. In the chaotic region the set of period-doubling cascading bifurcations into a chaotic state is observed. Beyond this regime there is no longer a bounded solution and turbulent behavior ensues, characterized by strange attractors in phase space. The structure of these strange attractors can be understood in terms of two distinct time scales which the motion exhibits. The shorter time scale corresponds to fast oscillations between a small numbers of wells; the longer time scale is associated with a slower diffusion throughout the periodic lattice.

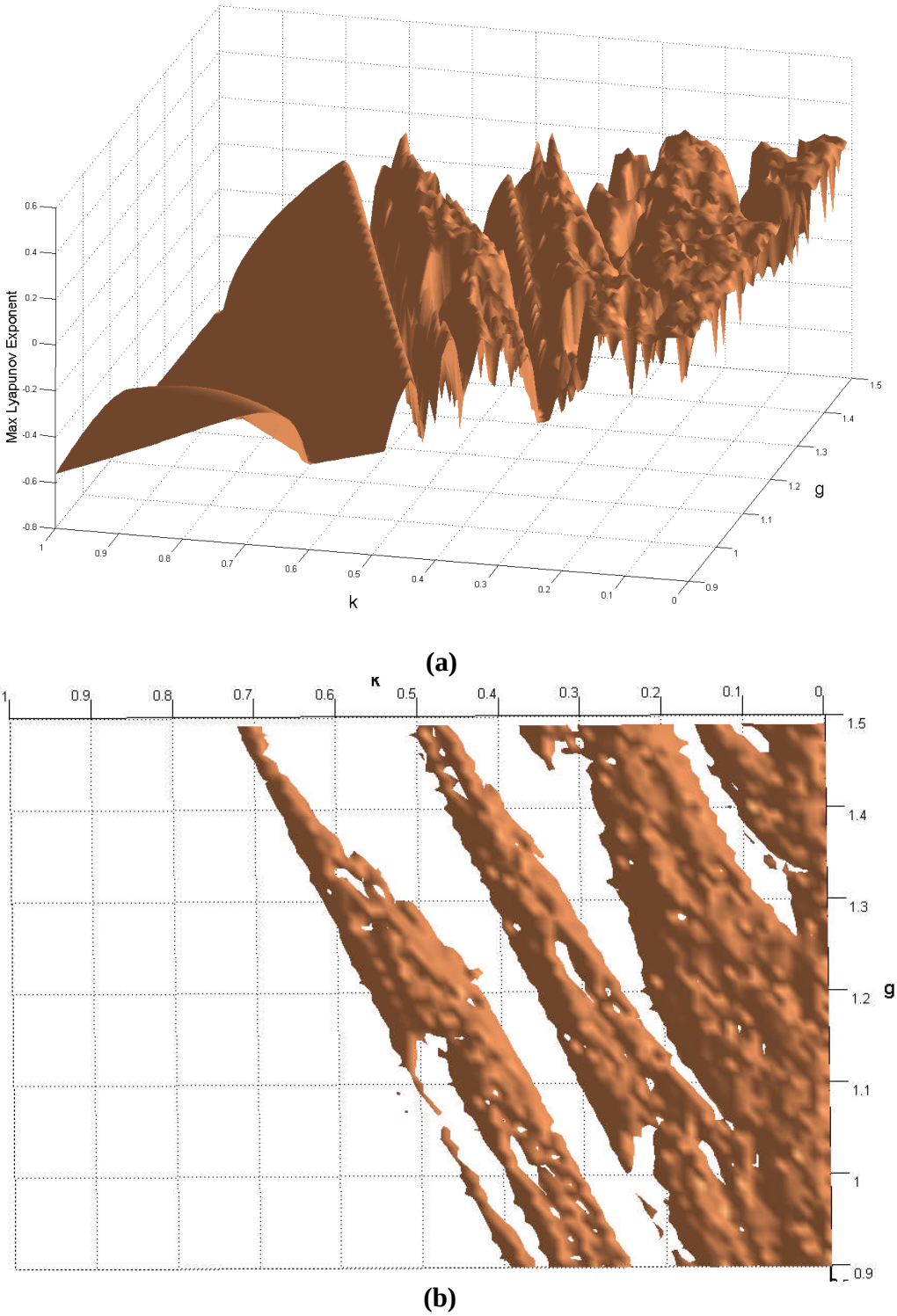
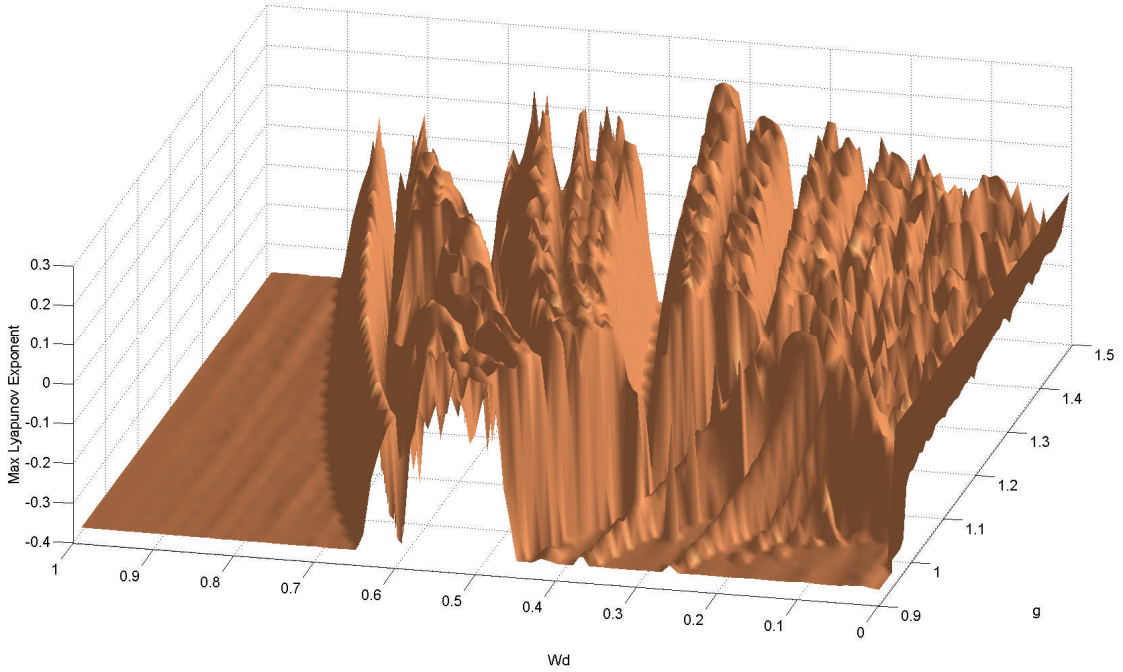
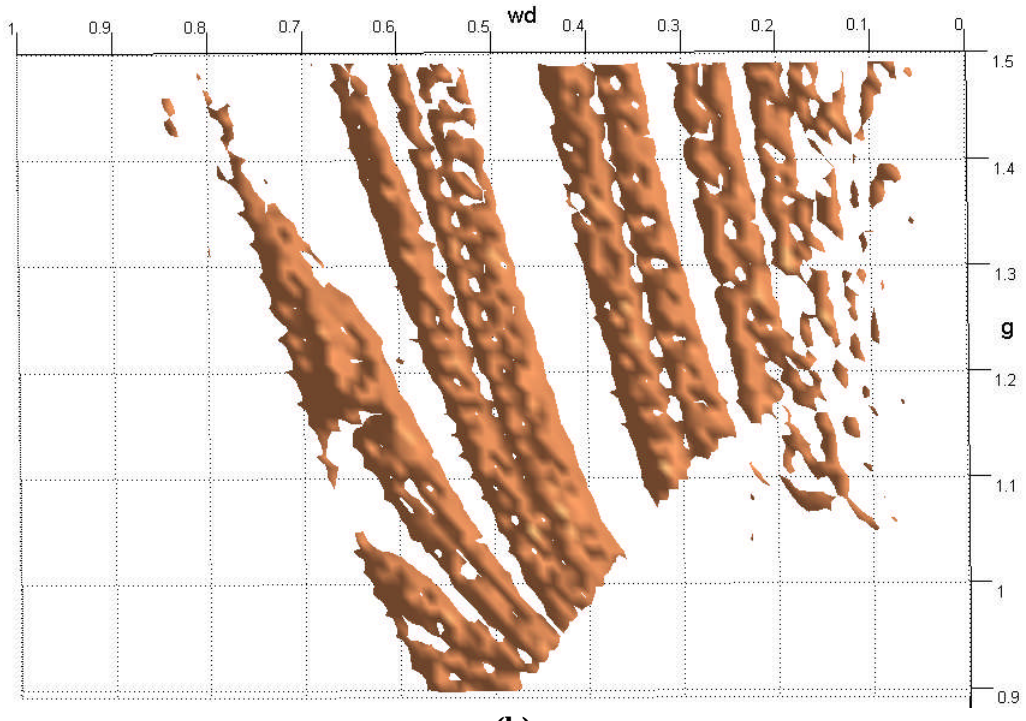


Figure 3.16 (a) First LE (λ_1) and (b) Josephson junction positive first LE (λ_1^+) for g vs. k. g: $0.9 \leq g \leq 1.5$, [Δg=0.01]; k: $0 \leq k \leq 0.9$, [Δk=0.01]; ω_d=2/3, 10^5 iterations, 10^4 transients, ε = 0.01, GSR steps = 5.

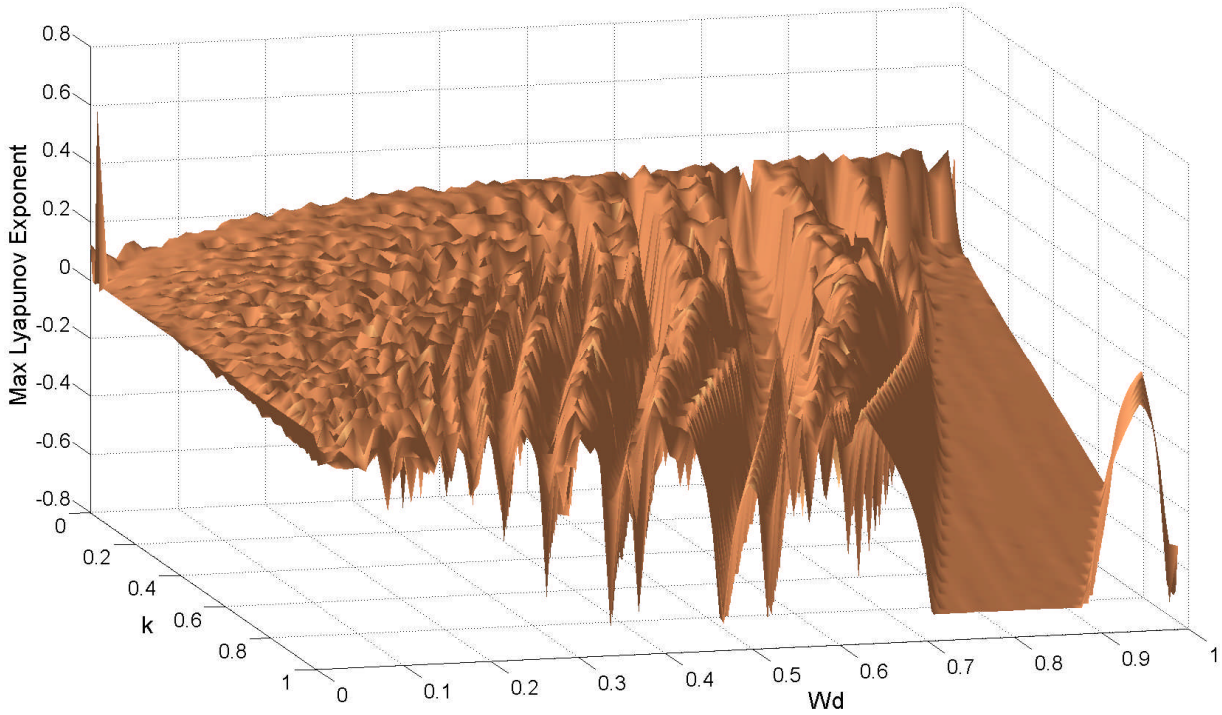


(a)

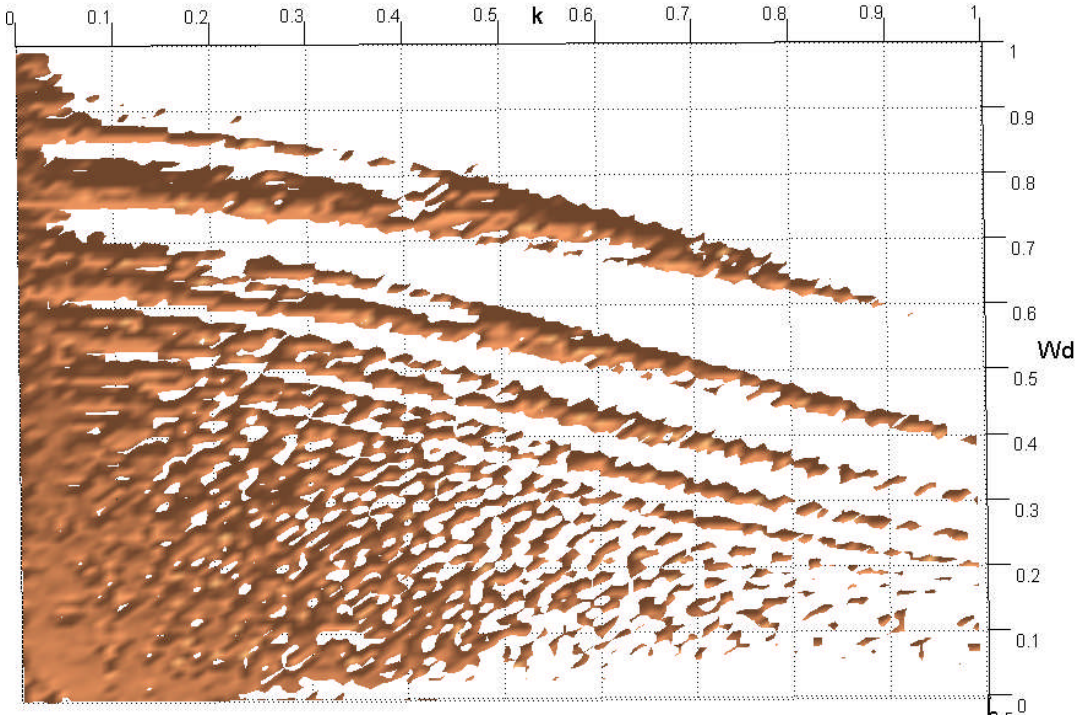


(b)

Figure 3.17 (a) Josephson junction system first LE (λ_1), (b) positive first LE (λ_1^+) for g vs. ω_d. g: $0.9 \leq g \leq 1.5$, Δg=0.01, k = 0.5, ω_d: $0 \leq \omega_d \leq 0.9$ [$\Delta\omega_d$ = 0.01], 10^5 iterations, 10^4 transients, ε = 0.01, GSR steps= 5.



(a)



(b)

Figure 3.18 (a) Josephson junction system first LE (λ_1), (b) positive first LE (λ_1^+) for k vs. ω_d. k: $0 \leq k \leq 1$ [$\Delta k = 0.01$]; ω_d: $0 \leq \omega_d \leq 1$ [$\Delta\omega_d = 0.01$]; g =3.8, 10^5 iterations, 10^4 transients, $\varepsilon = 0.01$, GSR steps = 5.

Our results suggest an explanation for the noise-rise phenomenon based on the intrinsic nonlinearity of the junction dynamics. Whether this theory accurately describes the observed behavior depends on the extent to which the driven damped oscillator models the actual junction dynamics. If that were the case, the phase diagrams in Figures 3.16, 3.17 and 3.18 could also provide some guideline for operating superconducting parametric amplifiers in noise-free regions. Moreover, if broad-band noise in Josephson oscillators is due to the presence of strange attractors in phase space, they could become likely candidates for the study of solid-state turbulence and nonlinear dynamics. Finally, we should point out that these results are of relevance to a wide variety of problems that can be modeled by the driven damped oscillator.

3.6 Lyapunov Fractal Dimension

Fractal dimension is a geometrical factor which can characterize the chaotic topology of strange attractor. This is the dimension of the topological structure which trajectories achieve by winding more and more densely without intersecting each other. For example, let M be the subset of the attractor in the n-dimensional phase space. If the phase space is covered by cubes with edge length ε, and $N(\varepsilon)$ be the number of cubes which contain a piece of attractor M. Then if ε contracts to zero, the limit of the ratio of the logarithm of $N(\varepsilon)$ to the logarithm of ε, that is $D = \lim_{\varepsilon \to 0} \{\log_2 N(\varepsilon) / \log_2(\varepsilon)\}$ is called the fractal dimension. This definition of fractal dimension is related to the values of Lyapunov exponents, where it is known as Kaplan-York fractal dimension. If the attractor is a point, the fractal dimension is zero. For a stable limit cycle the fractal dimension is one. But for chaotic systems the fractal dimension is not an integer value. In general, the fractal dimension can be calculated only numerically. One of the methods is introduced by Kaplan and Yorke [54]. They introduce a measure for the structural complexity of strange attractors using the Lyapunov exponent spectrum. In their proposition they relate the

attractor information dimension which is a static property to the Lyapunov exponent spectrum which is a dynamic property. In general, the Lyapunov dimension is given by

$$d_f = j + \sum_{i=1}^{j} \frac{\lambda_i}{|\lambda_{j+1}|},$$

$$\sum_{i=1}^{j} \lambda_i > 0,$$

$$\sum_{i=j+1}^{n} \lambda_i < 0.$$

j is the largest index that makes the sum of the successive Lyapunov exponents nonnegative, and then negative for $j+1$. For the same number of points in the time series, calculating d_f may be more accurate than calculating the morphological fractal dimension through the box-counting technique. In the simplest form, the Lyapunov fractal dimension of a 2-D map is given by $d_f = 1 + \frac{\lambda_1}{|\lambda_2|}, \lambda_1 < |\lambda_2|$, where the largest exponent λ_1 is positive and λ_2 is negative. For example, consider the Josephson junction system again. The Lyapunov exponents spectrum for following set of parameters, k=0.5, ω_d=2/3 and g = 1.16 expressed in base-2 is (λ_1, λ_2, λ_3)=(+, 0, −)=(0.24, 0, −0.86) bits/s. The Lyapunov sum is $\sum_{i=1}^{3} \lambda_i = 0.24 - 0.86 = -0.58$ which indicates that the dynamical system is dissipative. For the above spectrum, the largest integer j that still keeps the sum positive is j = 2, and λ_{j+1} = λ_3. Thus, the Lyapunov fractal dimension is $d_f = 2 + \frac{(0.24+0)}{|-0.86|} = 2.279$. This value of $d_f \approx 2$ is consistent with the flat shape of the Josephson attractor in Figure 3.18. Through same process by using the original parameters of $a = 1.4,\ b = 0.3$ we found the fractal dimension of 1.258 for the Henon map; again the value of $d_f \approx 1$ is consistent with the line-like shape of its strange attractor.

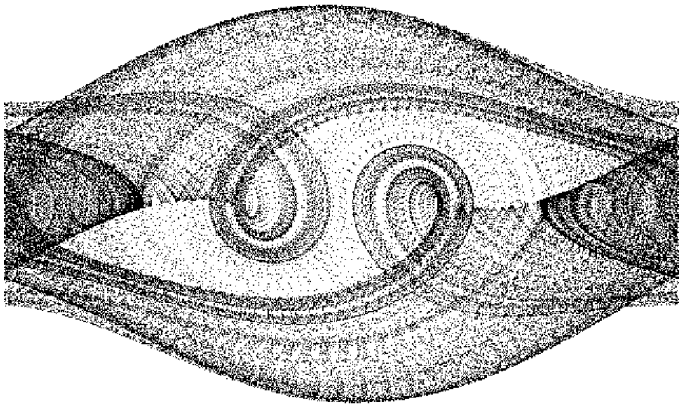


Figure 3.19 Josephson junction system strange attractor for g = 1.16, k = 0.5, ω_d = 2/3.

By knowing the Lyapunov exponents spectrum for a Josephson junction system we found the fractal dimension with a fixed step size of 0.001 for a microwave driving amplitude in the range of $0.9 \leq g \leq 1.5$. Dissipation coefficient and microwave source frequency are k = 0.5, ω_d = 2/3 respectively. The spectrum expressed in base-e along with the K-Y fractal dimension is shown in Figure 3.20. This indicates that Josephson junction system in chaotic regions has fractal structures.

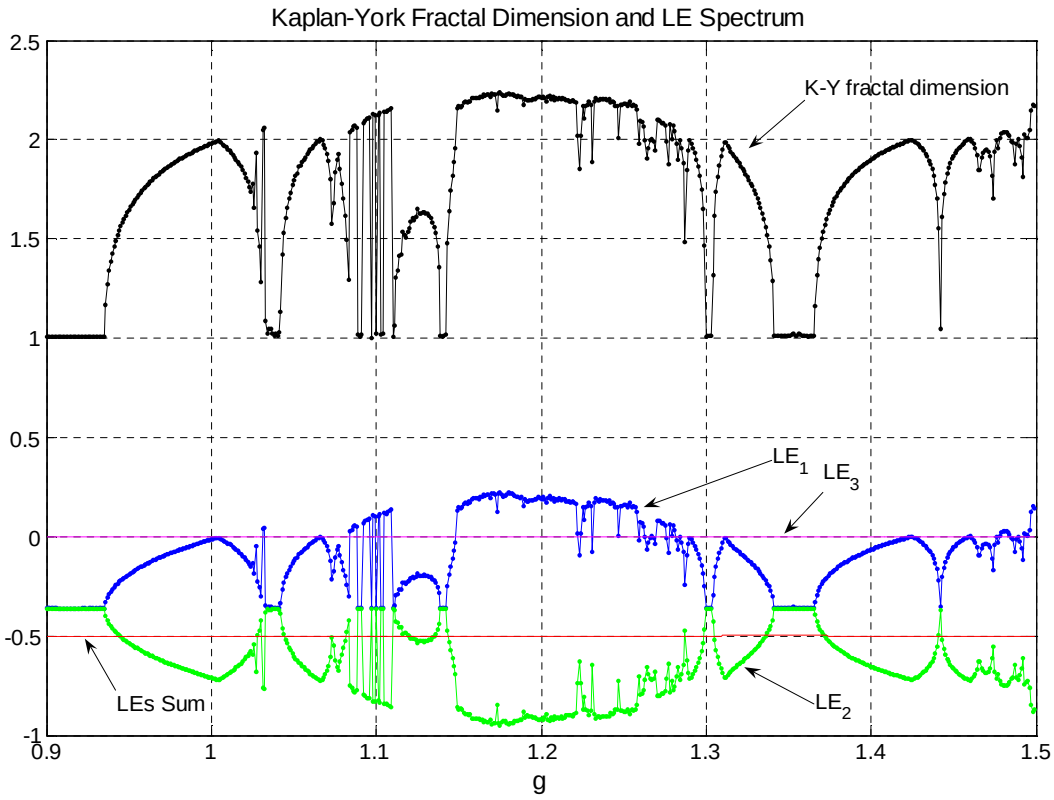


Figure 3.20 Josephson junction system Kaplan-York fractal dimension and LE spectrum (λ_1, λ_2, λ_3) and (ln2)$\sum\lambda_i$) vs. g: $0.9 \le g \le 1.5$ [$\Delta g = 0.01$], $k = 0.5$, $\omega_d = 2/3$, 10^5 iterations, 10^4 transients, $\varepsilon = 0.01$, GSR step =5.

3.7 Entropy Metrics

Similar to Lyapunov exponents, the Kolmogorov-Sinai entropy h_{K-S} is a measure of information production per iteration in maps or per unit of time in *n*- dimensional continuous dynamic systems. If the dynamics is preserving a measure *p*, then the Kolmogorov-Sinai expression $h_{K-S}(p)$ measures the asymptotic rate of creation of information by the dynamic system. It estimates the amount of randomness in the system which is not explained by the defining equations. Positivity of the Kolmogorov-Sinai entropy in general implies that at least one of the Lyapunov exponents of the motion is positive and chaos is observed. It is generally believed that the many body systems of statistical mechanics have positive Kolmogorov-Sinai entropies. KS entropy can be used to characterize chaos [55]. This is because non-chaotic

systems have $h_{K-S}=0$, while chaotic systems have $h_{K-S}>0$ and uncorrelated noise has $h_{K-S}=\infty$. The significance of the Kolmogorov-Sinai entropy is that it expands the static probabilistic Shannon entropy measure to dynamical systems which are dynamic and deterministic, but not static, and thus provides a continuous supply of new information as they evolve in chaos. The Kolmogorov-Sinai entropy was introduced by Kolmogorov in 1958 and developed further by Sinai, Ruelle, Grassberger, Procaccia and many other researchers.

There are several schemes to compute the Kolmogorov-Sinai entropy. If a dynamical system has several positive Lyapunov exponents, the following Ruelle inequality holds for most systems $h_{K-S}<\sum_{i=1}^{N}\lambda_i$, where *N* is the index of the smallest positive Lyapunov exponent. It has shown that the inequality also holds for dynamic flows [56]. Thus, the spectra of positive Lyapunov exponents provide a good estimate of h_{K-S} without reference to the source statistics. This is important because accurate estimates of the h_{K-S} entropy directly from the process statistics require a very large number of data points in a time series. For example, more than 10^6 iterations for the logistic equation and even more for higher dimensional dynamical systems are required. Kolmogorov-Sinai entropy can be generalized to $h_{K-S,q}$, with moment orders *q*. The $h_{K-S,q}$ is related to the Renyi *q*-order entropies h_q. Further, h_q is related to the Renyi fractal dimension spectrum d_q. This fractal dimension is more immune to noise than the entropy itself. These entropy-based metrics are important in the characterization and classification of chaos. To demonstrate this, consider Figure 3.21 which compares the Henon map bifurcation diagram to K-S entropy calculated by a box-counting algorithm. Parameter *a* is varied over a grid of 40 by 40 cells from 1 to 1.4, and *b* is 0.3. Observe the low entropy level in periodic windows. Also note

that there is a distinct difference between the 4-cycle periodic state and 8-cycle state K-S entropy level.

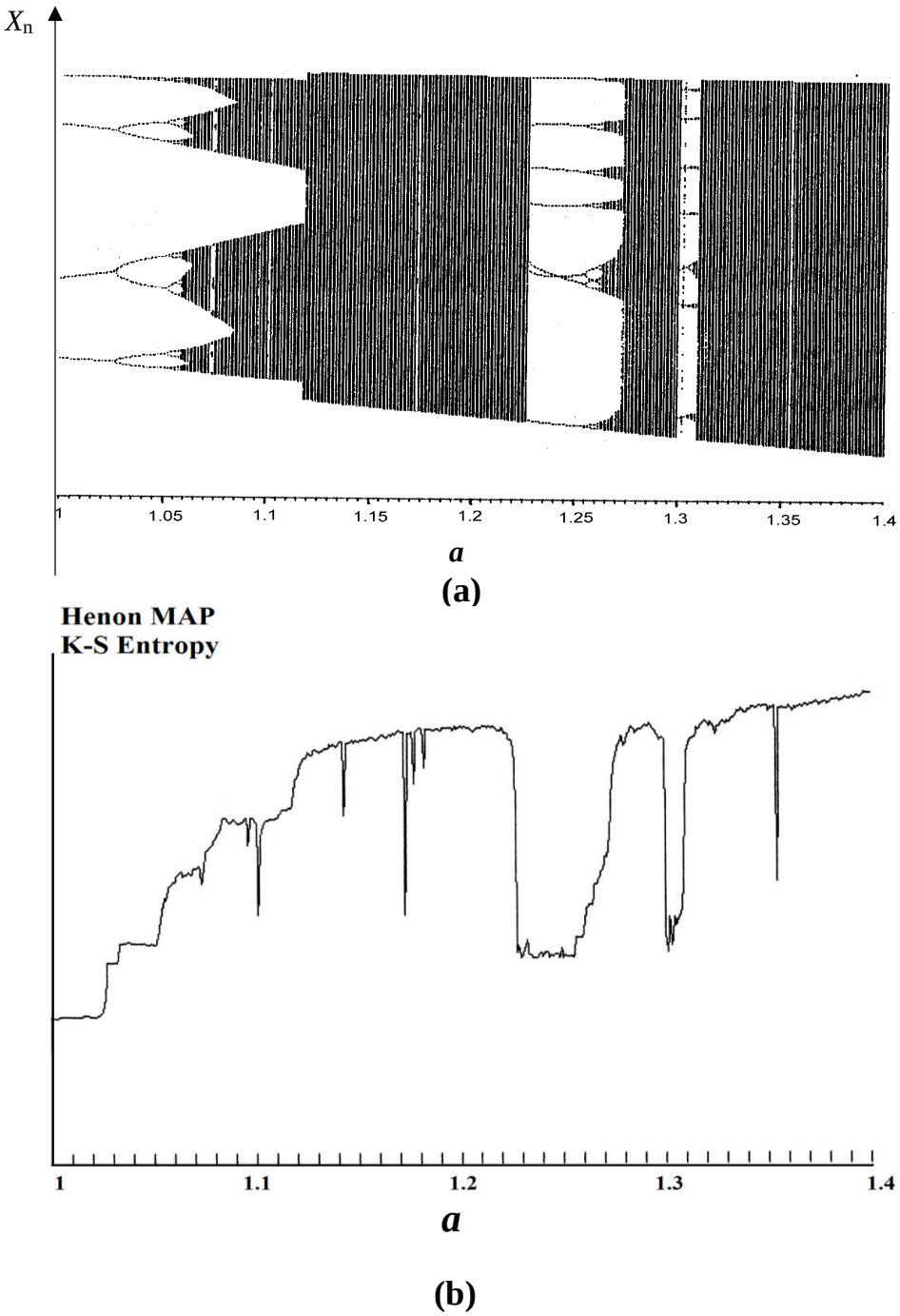


Figure 3.21 Henon map (a) bifurcation diagram (b) K-S entropy for $b = 0.3$, $1 \leq a \leq 1.4$ [$\Delta a = 0.01$].

CHAPTER 4

ALGORITHMIC COMPLEXITY

4.1 Introduction

In this chapter we review the algorithmic complexity concept. Then, an effective algorithm in computing the Lempel-Ziv complexity measure is introduced. The application to chaotic dynamical systems is discussed next.

In the mid 1960s, in the early stage of computer science, the general theory of Turing machines was well understood. Scientists were looking for a method to measure computation and information quantitatively. Algorithmic complexity was invented by R. J. Solomonov (1964), A. N. Kolmogorov (1965), and G. J. Chaitin (1969), independently and in this chronological order. This theory is now widely accepted as the standard approach that settled a half-century debate about the notion of randomness of an individual object. Kolmogorov (algorithmic) complexity has a plethora of applications in many areas including computer science, mathematics, physics, biology, and the social sciences. Intuitively, the amount of information in a finite string is the size (number of bits) of the smallest program that, started with a blank memory, computes the string and then terminates. A similar definition can be given for infinite strings, but in this case the program produces element after element forever. Such a definition would appear to make the amount of information in an object depend on the particular programming language used. It can be shown that all choices of universal programming languages lead to quantification of the amount of information that is invariant up to an additive constant.

The Kolmogorov complexity *K(S)* of a string *S* is the length of the shortest binary program (for a fixed reference universal programming language) that prints *S* as it's only output

and then stops. A string S is incompressible if *K(S)* is at least the length of *S*. The shortest way to describe *S* then is to give it literally. Similarly, a string *S* with length *L(S)* is nearly incompressible if *K(S)* is almost as large as *L(S)*. The appropriate standard for almost as large as above can depend on the context. Similarly, the conditional Kolmogorov complexity of S with respect to S' denoted by $K(S|S')$ is the length of the shortest binary program with extra information S' that prints *S*. A string S is incompressible relative to S' if $K(S|S')$ is large in the appropriate sense. Intuitively, we think of such patternless sequences as being random, and we use the term random sequence synonymously with incompressible sequence. The Kolmogorov complexity is a wonderful measure of randomness; however it is not computable, which obviously impedes some forms of practical use. Nevertheless, non-computability is not really an obstacle in a wide range of applications. As we will see in the next section, some measurable form of Kolmogorov complexity can be introduced in signal analysis.

4.2 Lempel-Ziv Complexity

This section, introduce and discuss the concept of Lempel-Ziv algorithmic complexity measure. Lempel-Ziv complexity (LZC) is a practical technique to characterize spatiotemporal patterns of nonlinear dynamical systems. This complexity measure is related to the number of distinct substrings (patterns) and the rate of their occurrence along a given sequence. It is an alternative tool for signal analysis in time domain. In recent years complexity measurement techniques have been applied to many signal analysis problems to evaluate the information content of finite sequences and in applications such as coding [57, 58, 59, 60], data compression [61, 62, 63, 64] and generation of test signals [65, 66, 67]. Extensive research has been carried out in recent years to apply LZ complexity in biomedical signal analysis as a metric to estimate

the information value of discrete-time physiologic signals [68]. In a recent study, a new sequence distance measure for phylogenetic tree construction has been proposed based on the relative information between the sequences using LZ complexity [69]. In brain function study LZ complexity is used to evaluate the brain information transmission [70]; as well as in electroencephalogram (EEG) complexity analysis in patients with Alzheimer's disease [71] and epileptic seizures [72]. In other studies, it is found that complexity is a good tool for monitoring the effect of anesthesia in patients. LZC is used as an indicator of changes in the brain during anesthesia. Researchers quantify the relationship between brain activity patterns and depth of anesthesia by analyzing the spatiotemporal patterns in the EEG's using Lempel-Ziv complexity analysis. They have recorded EEG during the procedure and patients' anesthesia states. According to the responsiveness component of the observer's assessment of alertness/sedation score, they assess the patients' anesthesia states. A score of zero or one is considered asleep and two or greater is considered awake. By estimating the LZC of the EEG signals, they are able to discriminate the performance of the brain regarding awake and asleep states. Compared with other measures, such as approximate entropy, spectral entropy, and median frequency, LZC demonstrates better performance across all of the patients [73, 74, 75]. In another study, the EEG complexity of ischemic cerebral injury is evaluated, and it is found that the EEG complexity is very sensitive to the extent of the injury [76]. Also it is used to quantify oscillations in uterine electromyography [77]. It is believed heart arrhythmias are a possible candidate for chaos in biological systems. This is an irregular beating of hearts in the form of arrhythmias and fibrillations. Therefore a measure of complexity has been applied to ECG signals to study heart dynamics. A simple system that exhibits an interesting irregular arrhythmia similar to that seen in humans consists of a rabbit heart septum preparation that is induced to beat chaotically

through perfusion with drugs such as a fast-acting version of digitalis, an overdose of which can cause arrhythmias. The chaotic beating of the rabbit heart tissue was converted to periodic beating through electrical stimuli applied at irregular timing dictated by chaos control [78, 79]. Other studies used complexity measures to detect ventricular tachycardia and fibrillation [80], [81]. In DNA studies LZC has been used for recognition of structural regularities [82], for complexity characterization of DNA sequences [83], to develop new methods for discovering patterns in DNA sequences by applying it to genomic sequences of Plasmodium falciparum [84] and to characterize the responses of neurons of the primary visual cortex to different kinds of stimuli [85]. In machine monitoring, the complexity of vibration signals from large rotating machinery is calculated. It is found that the Lempel-Ziv complexity is a useful measure for machine health evaluation and the complexity value could be correlated with insignificant variations in machine running conditions, which otherwise could not be captured using conventional parameters such as peak or RMS values [86]. As an example from behavior science, a complexity-based analysis of the turn-taking sequences that emerge in a Computer-Supported Collaborative Learning environment was studied to identify the underlying patterns in peers' turn-taking sequence by measuring the deviations in data complexity, and to reveal any possible changes due to the peers' feedback provision. Results show a clear transition to more complex turn-taking patterns when appropriate feedback is provided to the peers [87]. Last but not least, Lempel-Ziv complexity is an important measure used in cryptography [88]. For instance, it is used to test the randomness of the output of a symmetric cipher. In Chapter 9 we investigate the application of the Lempel-Ziv complexity measure in a performance analysis of best-rate channel coding algorithms with a chaotic mask for different generators.

4.3 Theoretical Background

Algorithmic complexity theory defines randomness based only on the characteristics of the signal, without any knowledge of the source of the data. Algorithmic complexity in one-dim to multi-dimensional discrete and continuous dynamical systems as a characterizing parameter is discussed next. We have used a logistic map, a Henon discrete map, a Lorenz system, a Josephson junction under microwave radiation system, a Josephson microbridge tetrode, and a hyperchaotic feedback control system to exemplify and illuminate the concept. We will see that for discrete maps as well as continuous nonlinear dynamical systems, an algorithmic complexity measure is as powerful as other characteristics like the spectrum of Lyapunov exponents and entropies. The numerical effort needed to extract this spectrum is rather large. This limits their determination to systems with dimensionality lower than ten. Therefore it is necessary to develop analytical tools in order to characterize chaotic motion in high-dimensional dynamical systems, e.g., spatiotemporal turbulence or poorly stirred chemical reactions.

The algorithmic complexity of a string is defined to be the length in bits of the shortest algorithm required for a computer to produce the given string. The shortest algorithms are referred to as minimal programs $H(S) \equiv \min_{U(P)=S} |P|$, where P is a program string used by a universal computer *U(P)* to produce the sequence *S*. The complexity of a string is thus the length in bits of the minimal program necessary to produce the given string. The definition of a random number can now be given as any binary string whose algorithmic complexity is judged to be essentially equal to the length of the string. Qualitatively, the information embodied in a random number cannot be reduced or compressed to a more compact form. For example consider the binary string $S = 0101010101\,0101\ldots0101$. A simple computer program to produce the given string might be: Print the string "01" ten times. To generate a string of "n", 01, where n is

large, the size of this program grows like $\log_2 n$. On the other hand for the digit string $S = 0101110000\ 1001010110\ 1010010100\ 0011011001\ 0$ almost the best one can do is: Print the string S. As the string S grows in length, the length of program grows like n, and the length of the computer program in bits is essentially the same as the length of S. Such a string satisfies the definition of a random number since the algorithmic complexity of the string is essentially the same as the length in bits of the string.

One of the major challenges in chaotic dynamics is to extract a meaningful signal from data that have every appearance of being random. Clearly algorithmic complexity is a concept aimed specifically at the problem of distinguishing between the random and the nonrandom. Nevertheless it can distinguish between chaotic, quasi-periodic, and periodic signals. The problem lies in determining a computable measure of complexity. No absolute measure is possible because minimal programs by definition correspond to random numbers, and it is not possible to determine a truly random number in any formal system. Nevertheless, it is possible to define a measure of complexity. A relative measure is sufficient for many purposes. For the first time Kasper and Schuster applied this idea to dynamic systems exhibiting chaos based on the work of Lempel and Ziv. The measure of complexity introduced by Lempel and Ziv is referred as LZ complexity for brevity (LZC). The LZ complexity measures the number of distinct patterns that must be copied to reproduce a given string. Therefore the only computer operations considered in constructing a string are copying old patterns and inserting new ones.

In general, transforming the signal to be analyzed into a data sequence whose elements are given by a few symbols performs LZ complexity analysis. Such a transformation is commonly referred to as a coarse-graining operation, which focuses on the intrinsic characteristics of the overall dynamics of the signals. That means in general that in any

dynamical system, it is possible to characterize the system behavior through its output responses by mapping the system output signal to a binary string and calculating the complexity measure of the time-series data. This approach does the classification of system behavior by the pattern formation in phase space attractors of the dynamical systems. Briefly described, a string S is scanned from left to right and a complexity counter $C(S)$ increased by one unit every time a new sub-string of consecutive digits is encountered in the scanning process. The resultant number $C(S)$ is the complexity measure of the string S. Clearly any procedure such as this will over estimate the complexity of strings, but nevertheless we expect comparisons to be meaningful. Complexity analysis based on the Lempel-Ziv definition is performed by transforming the signal to be analyzed into a data sequence whose elements are given by a few symbols. This is a transformation called a coarse-graining operation, which focuses on the intrinsic characteristics of the overall dynamics of the signal. Consider the sequence which is the output signal of a dynamic system. Then a threshold such as the mean value of the signal is first calculated. Next, a new sequence is reconstructed by comparing the value of each sample of the previous sequence with the mean value. If the value of the sample is larger than the threshold, set as 1, otherwise as 0. Thus, only two symbols exist in the new data sequence. This new data sequence is subsequently scanned from its first sample to the end. When a subsequence that is not encountered in the previous scanning process is discovered, the complexity value will be increased by one. Thus, the Lempel-Ziv complexity reflects upon the number of all different subsequences contained in the original data sequence. For purpose of generality, normalized complexity is often used to obtain a measure that is independent of the sequence length.

Algorithmic complexity theory defines randomness based only on the characteristics of the signal, without any knowledge of the source of the data. The algorithmic complexity of a

symbol string is defined to be the length in bits of the shortest algorithm required for a computer to produce the given string. The shortest algorithms are referred to as minimal programs. Any binary string whose algorithmic complexity is judged to be essentially equal to the length of the string is defined as a random sequence. Qualitatively, the information embodied in a random sequence cannot be reduced or compressed to a more compact form. As the string *S* grows in length, the length of the program grows like *n*, and the length of the computer program in bits is essentially the same as the length of *S*. Such a string satisfies the definition of a random sequence since the algorithmic complexity of the sequence is essentially the same as the length in bits of the sequence. Clearly algorithmic complexity is a concept aimed specifically at the problem of distinguishing between the random and the nonrandom. Nevertheless it can distinguish between chaotic, quasi-periodic, and periodic signals. The problem lies in determining a computable measure of complexity. No absolute measure is possible because minimal programs by definition correspond to random sequences, and it is not possible to determine a truly random sequence in any formal system because of the Gödel incompleteness theorem. However, it is possible to define a measure of complexity. A relative measure is sufficient for many purposes. The measure of complexity introduced by Lempel and Ziv is referred as LZ complexity or LZC for brevity. The LZ complexity measures the number of distinct patterns that must be copied to reproduce a given string. Therefore the only computer operations considered in reconstructing a string are inserting new patterns and copying old ones. More precisely, the LZ complexity of a given sequence *S* is the number of insertions of new symbols required to reconstruct *S*, where every attempt is made to construct *S* by copying alone without inserting any new symbols. The process is iterative and the first symbol must always be inserted. The minimum value for LZ complexity is two. Furthermore the LZ complexity measure for a given string S is unique and only relative

values of $C(n)$ are meaningful. In particular it is the comparison with the complexity of the random string that is meaningful. Thus, one should always compare the LZ complexity of a given string to the LZ complexity of random strings of the same length, $\lim_{n\to\infty} C(n)/b(n)$ where for a random string of length n, the LZ complexity is given by $b(n) = n/\log_2 n$. In quantifying these ideas it becomes necessary to introduce certain definitions. Let $\hat{A}$ denote the alphabet of symbols from which the finite length sequence S is constructed. If there are two symbols in the alphabet, they are called bits. $L(S)$ represents the length of this sequence. A sequence S may be written in the form $S = s_1 s_2 \cdots s_r s_{r+1} \cdots s_n$, where symbol s_r is mapped from phase space attractor pattern of a dynamic system based on a mapping rule. The vocabulary of a sequence S denoted by $V(S)$ is the set of all subsequences of S. For example if $\hat{A} = \{0,1\}$ and $S = 0010$, the vocabulary of a sequence S denoted by $V(S)$ is the set of all subsequences of S. Thus, the vocabulary of sequence S is given by $V(S)$={0, 1, 00, 01, 10, 001, 010, 0010}. One can consider $V(S)$ as all the possible words that can be constructed from different bit combinations (zeros and ones) in the order they appear in S. The operator " ^ ", eliminates the last bit from sequence. Thus, the notation $S^{\wedge}$ denotes the string $s_1 s_2 \cdots s_{n-1}$ which is obtained from S by eliminating the last symbol: $S = s_1 s_2 \cdots s_r s_{r+1} \cdots s_n \rightarrow S^{\wedge} = s_1 s_2 \ldots s_{n-1}$. Also the symbol " \ " denotes the end of each different subsequence. In other words in the sequence $s_1 s_2 \cdots s_j \backslash$, the last bit ($s_j$) is newly inserted. Conceptually in order to calculate the LZ complexity, we start from the first symbol in string S and try to reconstruct this sequence with two operations of INSERT new pattern and COPY repeated patterns. S' is a reconstructed sequence at the j^{th} step of reading from buffer sequence S. The current subsequence Q, acts as a sliding window for scanning the buffer sequence S. The concatenated subsequence is SQ and the previous subsequence is $SQ^{\wedge}$. A decision is made for

the question asking whether Q belongs to the vocabulary of the previous sequence $\{Q \in V(SQ\wedge)\}?$; in this way we determine whether the scanned subsequence is a new bit pattern or a repeated pattern. Hence we simply scan and compare the symbol patterns in the dictionary of each new word in the string. If the symbol pattern is a new pattern we INSERT that word into the dictionary (LZ complexity is raised by one unit), if it is not a new bit pattern it is simply COPY (LZ complexity does not change). The total number of inserted words into the dictionary is actually the Lempel-Ziv complexity. The complexity analysis procedure at the j^{th} step of the reconstruction process is as follows.

Step j) The buffer sequence is $S = s_1 s_2 \cdots s_j s_{j+1} \cdots s_n$. Assume that the j^{th} character, s_j has just been inserted into the scanning process. The reconstructed sequence is $S' = s_1 s_2 \cdots s_j \backslash$; the current subsequence $Q = s_{j+1}$, and the concatenated subsequence is $S'Q = s_1 s_2 \cdots s_j s_{j+1}$, then the previous subsequence is $S'Q\wedge = s_1 s_2 \cdots s_j$. If the vocabulary, $Q \in V(S'Q\wedge)$ contains $Q = s_{j+1}$, i.e. $Q \in V(S'Q\wedge)$, it means Q is not a new subsequence. Then proceed to next step by choosing $Q = s_{j+1} s_{j+2}$, at this time LZC is not increased. But if the vocabulary, $V(S'Q\wedge)$ does not contain $Q = s_{j+1}$, i.e., $Q \notin V(S'Q\wedge)$, Q is a new subsequence and should be included in dictionary. This new subsequence increases LZC by one unit. Then proceed to the next step by choosing $Q = s_{j+2}$ This is a repetitive process and continues until the last bit in the sequence S is analyzed. Let us apply the LZC analysis to a few examples.

Example 1 LZ Complexity of sequence the $S = s_1 s_2 \cdots s_j s_{j+1} \cdots s_n = 11111 \cdots 11$, with length $L(S) = n$, is determined as follows. Let S' be the reconstructed sequence of S.

Step 1) The first character is $s_1 = 1$ and always is a new character. Therefore, the first subsequence is $S' = s_1 \backslash = 1 \backslash$.

Step 2) The second character is $s_2 = 1$ and this is identical to the first subsequence. The current subsequence is $Q = s_2 = 1$, the reconstructed sequence is $S' = 1 \backslash$, concatenated subsequence $S = s_1 s_2 = 11$, and previous subsequence $S'Q\hat{} = 1$. Since Q already exist in vocabulary, i.e., $(Q = 1) \in V(S'Q\hat{}) = \{1\}$, therefore Q is not a new subsequence.

This process is repeated up to the last bit. Finally the reconstructed sequence becomes $S' = Insert(1) \backslash Copy(1) = 1 \backslash 111 \cdots 1$. By definition, LZC is equal to number of parts in S' separated by \, thus $C(S) = 2$.

Example 2 LZ Complexity of the sequence $S = s_1 s_2 \cdots s_j s_{j+1} \cdots s_n = 01010101 \cdots 0101$ is $C(S) = 3$, since S can be reconstructed as

$S' = Insert(0) \backslash Insert(1) \backslash Copy(01) = 0 \backslash 1 \backslash 010101 \cdots 01$.

Example 3 Here we explicitly apply the algorithm to the following sequence with length 12:

$S = s_1 s_2 \cdots s_j s_{j+1} \cdots s_{12} = 0011110000\ 11$.

Step 1) The first character is $s_1 = 0$ and always is a new character. Therefore, the first subsequence is $S' = s_1 \backslash = 0 \backslash$.

Step 2) Second character is $s_2 = 0$ and this is identical to the first subsequence. The current subsequence is $Q = s_2 = 0$, the reconstructed sequence is $S' = 0 \backslash$, the concatenated subsequence is $S'Q = s_1 s_2 = 00$, and the previous subsequence is $S'Q\hat{} = 0$. Therefore $Q = s_2 = 0$ is not a new subsequence since $(Q = 0) \in V(S'Q\hat{}) = \{0\}$.

Step 3) The third character is $s_3 = 1$. The reconstructed sequence is $S' = 0\backslash$, the current subsequence $Q = 01$, the concatenated subsequence $S'Q = s_1 s_2 s_3 = 001$ and the previous subsequence $S'Q\hat{} = 00$. Since $(Q = 01) \notin V(S'Q\hat{}) = \{0, 00\}$, $Q = 01$ is a new subsequence.

Step 4) The fourth character is $s_4 = 1$. The reconstructed sequence $S' = 0\backslash 01\backslash$, the current subsequence $Q = s_4 = 1$, the concatenated subsequence $S'Q = 0011$ and previous subsequence $S'Q\hat{} = 001$. Since $(Q = 1) \in V(S'Q\hat{}) = \{0, 1, 00, 01, 001\}$, therefore, the dictionary already contains $Q = s_4 = 1$, so is not a new subsequence .

Step 5) The fifth character is $s_5 = 1$. Since reconstructed sequence $S' = 0\backslash 01\backslash$, the current subsequence $Q = 11$ and the concatenated subsequence $S'Q = 00111$, then the previous subsequence is $S'Q\hat{} = 0011$. Since $(Q = 11) \in V(S'Q\hat{}) = \{0, 1, 00, 01, 11, 001, 011, 0011\}$, therefore, the dictionary already contains it, so is not a new subsequence .

Step 6) The sixth character is $s_6 = 1$. The reconstructed sequence is $S' = 0\backslash 01\backslash$. The current subsequence is $Q = 111$, the concatenated subsequence $S'Q = 001111$ and the previous subsequence is $S'Q\hat{} = 00111$. Since,

$(Q = 111) \in V(S'Q\hat{}) = \{0, 1, 00, 01, 11, 001, 011, 111, 0011, 00111\}$, therefore, the dictionary already contains it, so is not a new subsequence.

Step 7) The seventh character is $s_7 = 0$. The reconstructed sequence is $S' = 0\backslash 01\backslash$, the current subsequence $Q = 1110$, the concatenated subsequence $S'Q = 0011110$ and the previous subsequence is $S'Q\hat{} = 001111$. Since,

$(Q = 1110) \notin V(S'Q\hat{}) = \{0, 1, 00, 01, 11, 001, 011, 111, 0011, 0111, 1111, 00111, 01111, 001111\}$,

therefore the dictionary does not contain it, so is a new subsequence.

Step 8) The eighth character is $s_8 = 0$. The reconstructed sequence is $S' = 0 \backslash 01 \backslash 1110 \backslash$, the current subsequence is $Q = 0$, the concatenated subsequence $S'Q = 00111100$ and the previous subsequence is $S'Q^{\wedge} = 0011110$. Since

$$(Q = 0) \in V(S'Q^{\wedge}) = \left\{ \begin{matrix} 0,1,00,01,10,11,001,011,110,111,0011,0111,1110,1111,00111,01111,11110,001111, \\ 011110,0011110 \end{matrix} \right\} \text{ the}$$

dictionary contains it, so $Q = 0$ is not a new subsequence.

Step 9) The ninth character is $s_9 = 0$. The reconstructed sequence is $S' = 0 \backslash 01 \backslash 1110 \backslash$. The current subsequence is $Q = 00$, the concatenated subsequence $S'Q = 001111000$ and the previous $S'Q^{\wedge} = 00111100$. Since

$$(Q = 00) \in V(S'Q^{\wedge}) = \left\{ \begin{matrix} 0,1,00,01,10,11,000,001,011,110,111,1000,0011,0111,1110,1111,11000, \\ 00111,01111,11110,11100,001111,011110,111100,0111100,0011110,00111100 \end{matrix} \right\},$$

the dictionary contains it, so is not a new subsequence.

Step 10) The tenth character is $s_{10} = 0$. The reconstructed sequence $S' = 0 \backslash 01 \backslash 1110 \backslash$. The current subsequence $Q = 000$, concatenated subsequence $S'Q = 0011110000$ and previous subsequence is $S'Q^{\wedge} = 001111000$. Since

$$(Q = 000) \in V(S'Q^{\wedge}) = \left\{ \begin{matrix} 0,1,00,01,10,11,000,001,011,110,111,1000,0011,0111,1110,1111,11000, \\ 00111,01111,11110,11100,001111,011110,111100,0111100,0011110,00111100,\ldots 001111000 \end{matrix} \right\},$$

the dictionary contains it, so is not a new subsequence.

Step 11) The eleventh character is $s_{11} = 1$. The reconstructed sequence $S' = 0 \backslash 01 \backslash 1110 \backslash$. The current subsequence $Q = 0001$, the concatenated subsequence $S'Q = 00111100001$ and the previous subsequence is $S'Q^{\wedge} = 0011110000$. Since

$$(Q=0001)\notin V(S'Q^{\wedge})=$$
$$\left\{\begin{array}{l}0,1,00,01,10,11,000,001,011,110,111,1000,0011,0111,1110,1111,11000,\\ 00111,01111,11110,11100,001111,011110,111100,0111100,0011110,00111100,\ldots 001111000,\\ \ldots,0011110000\end{array}\right\},$$

Thus the dictionary does not contain $Q=0001$, so it is a new subsequence.

Step 12) The twelfth character is $s_{12}=1$. The reconstructed sequence $S'=0\backslash 01\backslash 1110\backslash 0001\backslash$. The current subsequence $Q=1$, concatenated subsequence $S'Q=001111000011$ and previous subsequence is $S'Q^{\wedge}=00111100001$. Since

$$(Q=1)\in V(S'Q^{\wedge})=$$
$$\left\{\begin{array}{l}0,1,00,01,10,11,000,001,011,110,111,0001,1000,0011,0111,1110,1111,11000,\\ 00111,01111,11110,11100,001111,011110,111100,0111100,0011110,00111100,\ldots 001111000,\\ \ldots,0011110000,\ldots,00111100001\end{array}\right\},$$

the dictionary contains $Q=1$, so is not a new subsequence.

By this process the, sequence is scanned and segmented as follows $S'=0\backslash 01\backslash 1110\backslash 0001\backslash 1$. The number of distinct patterns of the string S' that are separated by \, is equal to the LZ complexity of S, therefore $C(S) = 5$.

Figure 4.1 shows the LZ complexity calculation algorithm flowchart. At the initialization step, the buffer is set to zero, the complexity counter LZC is cleared to one, and character index L is set to one. In next step, a character is read from the string, starting from the first bit, K =1. Next, the bit is stored into the buffer and concatenated with the subsequence already in the buffer. Then the bit character, proceeding, is stored into buffer and concatenated with the subsequence in the buffer. The current subsequence located in the buffer is compared to the previous subsequence located in the buffer. If the previous subsequence does contain the current subsequence (i.e., the current subsequence can be copied from the previous one), then the current subsequence is not a new sequence and the algorithm just updates the character index for reading

the next character at the next step (step 200). If the previous subsequence does not contain the current subsequence, then the current subsequence is a new one and the new subsequence is marked and the complexity counter is increased by one and buffer is set to zero. Then the process continues to update the character index. The current index is compared to the length of the sequence. If it is not larger than the sequence length, then a new character is read (step 100). If it is larger than the length of the sequence, the scanning process is terminated (step 300).

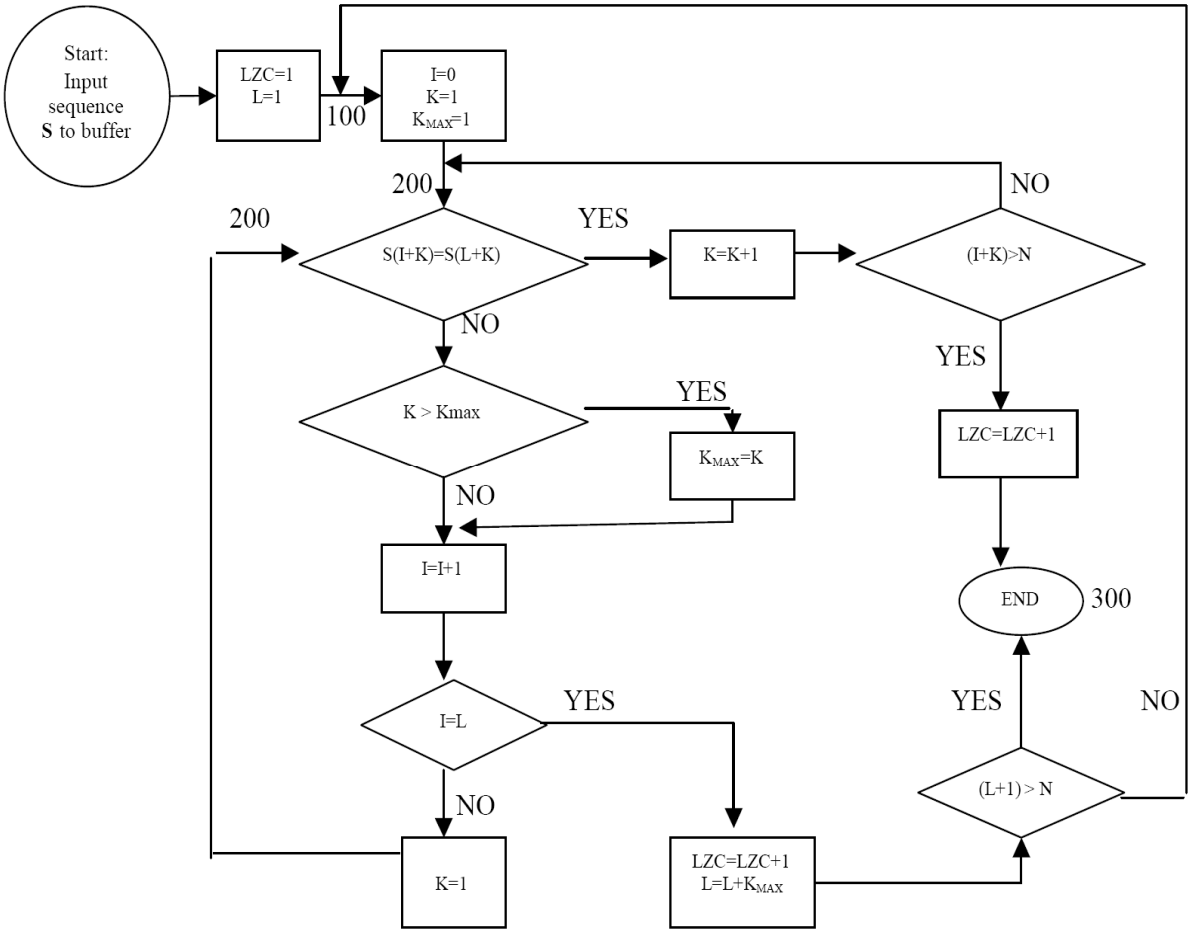


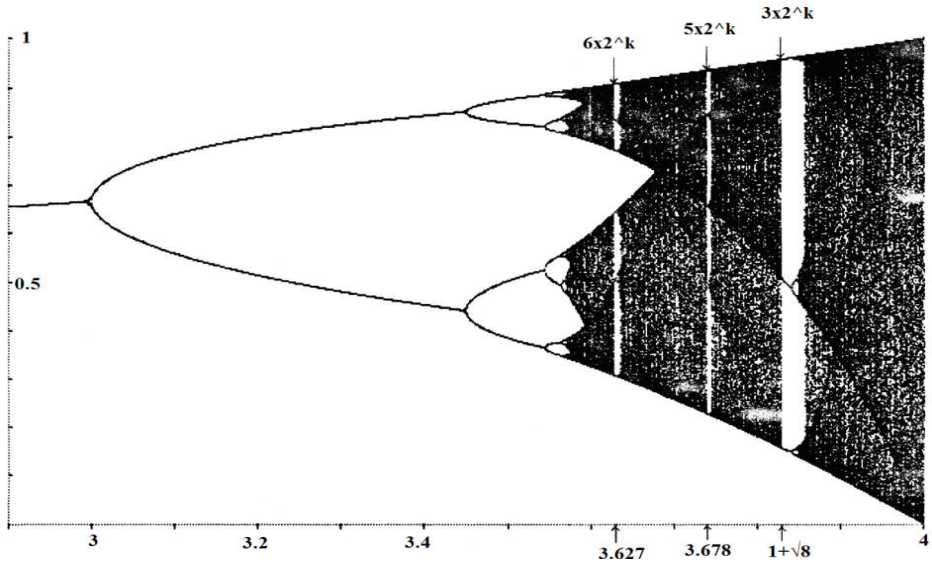
Figure 4.1 LZ complexity analysis flowchart for sequence *S* of length *N*.

4.4 Nonlinear Dynamic Systems Complexity Analysis

One of the major challenges in chaotic dynamics is to extract a meaningful signal from data that have every appearance of being random. In order to distinguish order from disorder a Lempel-Ziv complexity measure is used as a characteristic metric in discrete and continuous dynamic systems. For the first time Kasper and Schuster applied this idea to dynamic systems exhibiting chaos [19]. Our efforts here are directed toward outlining a computational algorithm in complexity analysis for various dynamic systems. To exemplify and illuminate the concept we apply the LZ complexity analysis to discrete maps like logistic and Henon maps as well as continuous systems such as a single Josephson junction and microbridge device models, quasi-periodic feedback controlled systems, coupled oscillators and the quasi periodic Fibonacci system. The general approach is that the phase space pattern of a certain nonlinear system is mapped to an array of symbols. The resulting sequence is stored in a buffer. Then the LZ complexity of the sequence is estimated. The process of mapping for each system can be different and will be explained and justified. Also the results are compared to the Lyapunov exponent measure of the same system. It is observed that for a nonlinear dynamical system the LZ complexity measure is as powerful as other characteristics like the Lyapunov spectrum. The numerical effort needed to extract these spectra is rather large. This limits their determination to systems with dimensionality lower than ten. Therefore it is necessary to develop and implement analytical tools like LZC in order to characterize chaotic motion in high-dimensional dynamical systems, like spatiotemporal turbulence, or poorly stirred chemical reactions. Comparing LZC to LE metrics one notices that both metrics are capable of distinguishing chaos from order but only the complexity metric is capable of distinguishing deterministic chaos from random noise.

4.4.1 Logistic Map LZ Complexity

In this section we study the finite strings generated by a logistic map. We start by using the partition two. The string S, is generated by taking 0 if $X(n) < 0.5$, and 1 otherwise. We find the LZC for a logistic map by analyzing sequences of length 10000. This result is shown in Fig. 4.2. In the case $\mu = 4$ we find fully developed chaos with $C(n) = 772$. Three different behaviors for this system can be distinguished. The LZC has a finite value for large string length. All marginally stable solutions including $\mu = \mu_\infty$ have zero Lyapunov exponents, but the LZC is significant when $\mu \to \mu_\infty$. This shows that LZC is a measure which reflects the dynamics variety of the system in a more precise manner in characterizing order or disorder.



(a)

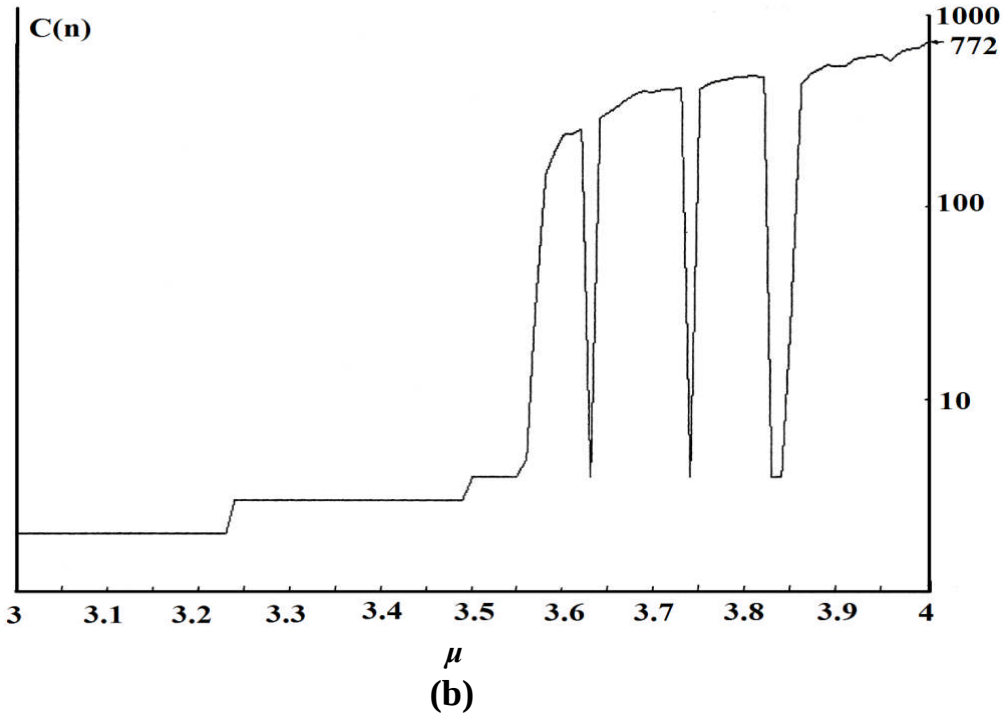


Figure 4.2 (a) Logistic map bifurcation (b) LZC when $3 \le \mu \le 4$, $1 \le a \le 1.4$ [$\Delta\mu = 0.01$].

4.4.2 Henon Map LZ Complexity

The Henon map is a prototypical two-dimensional invertible iterated map with chaotic solutions. This map is defined by the following quadratic nonlinear recursive map

$$H:\begin{cases} X_{n+1} = 1 - aX_n^2 + Y_n \\ Y_{n+1} = bX_n \end{cases}$$

where $0 \le a \le 2$ and $|b| < 1$. The parameter b is a measure of the rate of area contraction (dissipation), and the Henon map is the most general two-dimensional quadratic map with the property that the contraction is independent of X and Y. The LZ complexity of the Henon map is examined by establishing decision criteria with an alphabet for constructing strings. The decision rule is as follows. The Henon map phase space is divided into four regions: ($-X$, $-Y$), (X, Y), ($-X$, Y), (X, $-Y$) as shown in Figure 4.3. When the dynamical response of the system (X_n, Y_n), for a

given control parameter, lies in first and third quadrant of phase space, digit 1 is mapped to the string, otherwise digit 0 is inserted. The source entropy for this mapping scheme, $h = -[p\log_2 p + (1-p)\log_2(1-p)]$, where p is the probability that the dynamical response lies in the first and third quadrant of the phase space, is calculated. We find that for the Henon chaotic attractor the source entropy is close to unity, which justifies our choice of mapping scheme. We regard 10,000 points on the attractor as sufficient for measuring the complexity of two-dimensional attractors generated by the Henon map. Each array with length $n = 10000$, corresponds to 10000 iterations stored after transient values are ignored. Each experiment is performed for a specific value of control parameters. a is varied in the range [1, 4] and increased by step size $\Delta a = 0.001$. b is kept constant at 0.3 during the simulation.

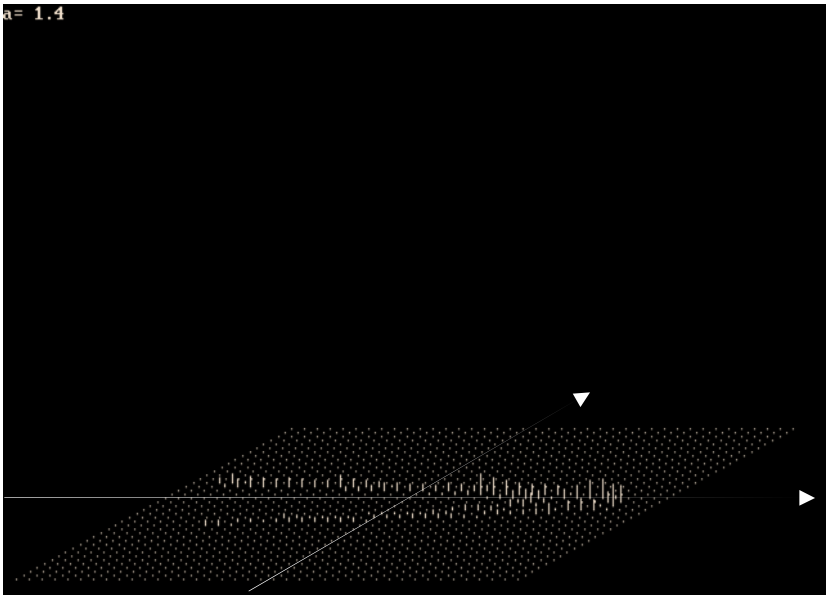


Figure 4.3 Henon map phase space (X_n, Y_n) for b=0.3, a=1.4.

Figure 4.4 shows the LZ complexity of the Henon map versus the control parameter a. Furthermore the bifurcation diagram is plotted on the same range. It is clear that when the Henon system has periodic behaviors of 1-cycle, 2-cycle…, that is, inside the bifurcation windows, the

complexity measure is small compared to the situations when the system behavior is chaotic. The results indicate distinct regions corresponding to chaotic and periodic behaviors. For values above a = 1.23 and less than 1.26, $C(n)$ shows a wide minimum at the windows. Lyapunov exponent, as expected from bifurcation diagram, reaches its largest possible value at $a = 1.4$ where the Lyapunov exponent of the Henon map has its maximum. This means that two different kinds of behavior of $C(n)$ have been found for the Henon map. For periodic orbits, $C(n)$ reaches a finite value for large n (at the theoretical limit $n \to \infty$, it goes to zero). At period doubling windows, $C(n)$ diverges with n, but the normalized complexity $C(n)/b(n)$, i.e., the complexity per digit, approaches to zero. For chaotic orbits the normalized complexity $C(n)/b(n)$ asymptotically reaches a finite positive value less than one. For example, when a = 1.4, $C(n)/b(n)$ is 0.74. This means that the complexity $C(n)$ is a more precise measure than the Lyapunov exponent for characterizing order or disorder.

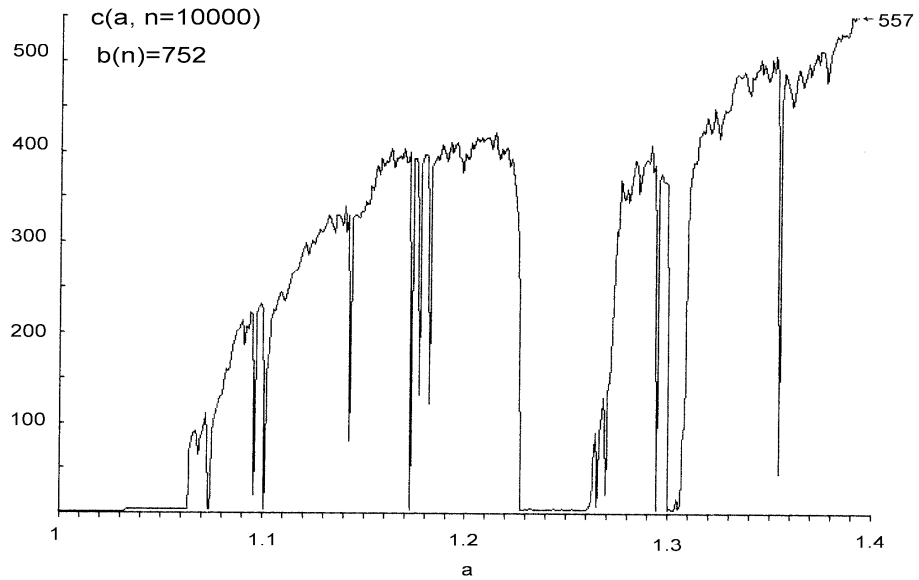


Figure 4.4 LZ complexity of strings constructed from the Henon map b = 0.3, $1 \leq a \leq 1.4$ [Δa = 0.001].

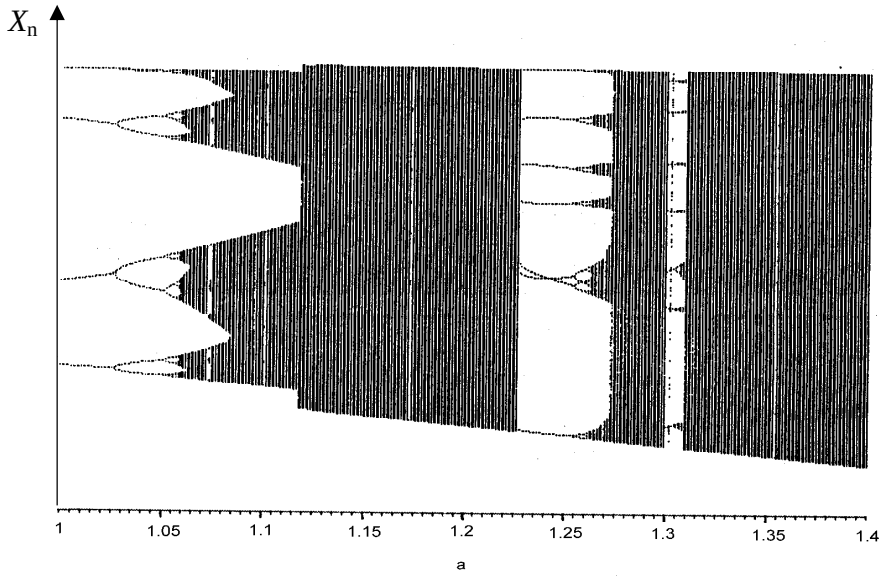


Figure 4.5 Henon map bifurcation diagram $b = 0.3$, $1 \leq a \leq 1.4$ [$\Delta a = 0.001$].

4.4.3 Josephson Junction LZ Complexity

The complexity of the oscillator map is calculated by setting control parameters to values $\omega_d = 2/3$ and $k = 0.5$. g is varied from 0.9 to 1.5 with increment 0.001. An array from 20000 iterations is constructed after removing the transient values for the specific external force amplitude (g). A digit 1 is mapped into the array if the voltage of the Josephson oscillator is larger than zero, otherwise digit 0 is inserted in the array. The Josephson single junction oscillator strange attractor and its Poincare map for g =1.2, k=0.5 and ω_d=2/3 is shown in Figure 4.6. The choice of using this encoding scheme is based on the symmetrical velocity distribution function. This function is calculated from a distribution density of 8×10^4 phase space points of a chaotic attractor (Figure 4.7), with applied normalized frequency of 2/3, dissipation coefficient of 0.5, and external driving amplitude of 1.2. We have computed the algorithmic complexity measure of each experiment versus the corresponding driving amplitudes as shown in Figure 4.8. For comparison purposes we include the bifurcation diagram (Figure 4.9). The results indicate

distinct regions corresponding to chaotic and periodic behaviors. When oscillator system has periodic behaviors of 1-cycle, 2-cycle, … as displayed in bifurcation windows, the LZ complexity measure is small. This corresponds to the limit-cycle trajectory or an attracting 2-torus. On the other hand the chaotic states with some trajectories expanding, while others are contracting, have large LZ complexity; this is nicely mirrored by the increasing values of the LZC plot.

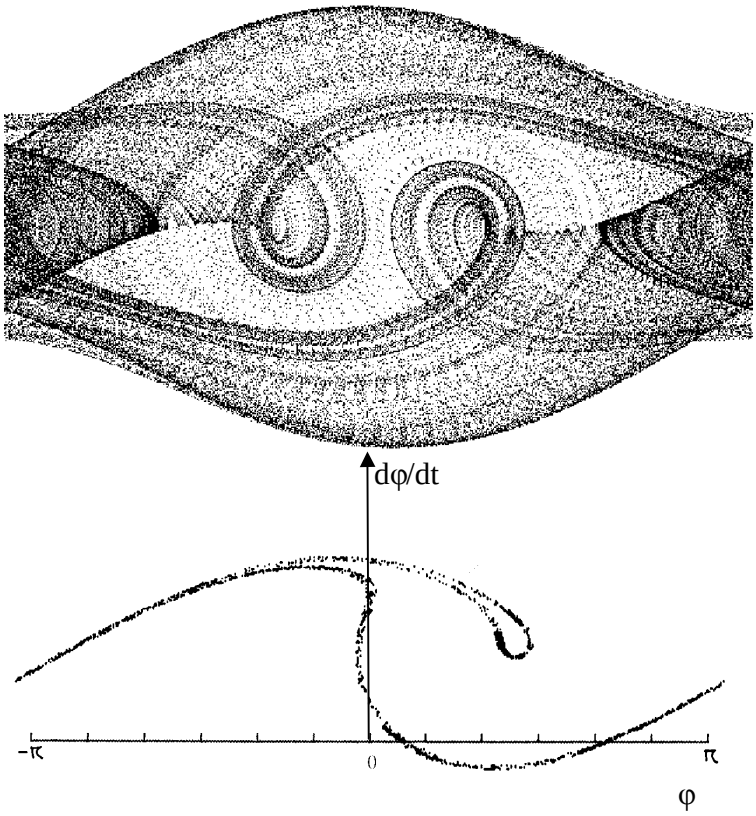


Figure 4.6 Josephson junction system strange attractor and its Poincare section $g = 1.2$, $k = 0.5$, $\omega_d = 2/3$.

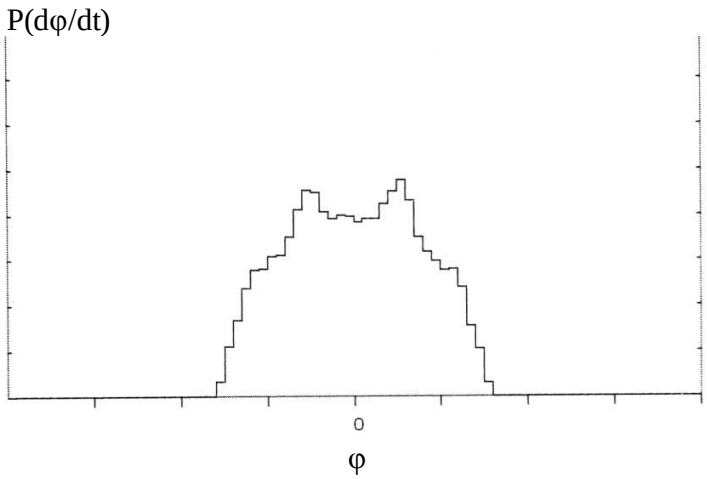


Figure 4.7 Josephson junction oscillation strange attractor probability density function g = 1.2, k = 0.5, ω_d = 2/3.

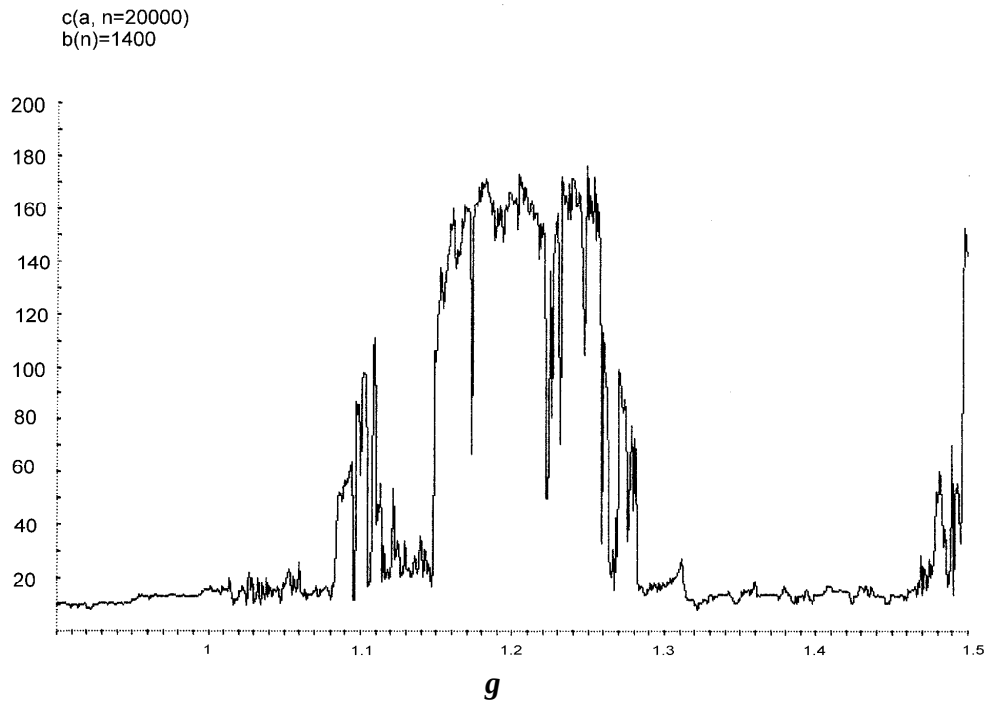


Figure 4.8 Josephson junction system LZ complexity measure vs. microwave amplitude g [Δg = 0.001]; k = 0.5, ω_d = 2/3.

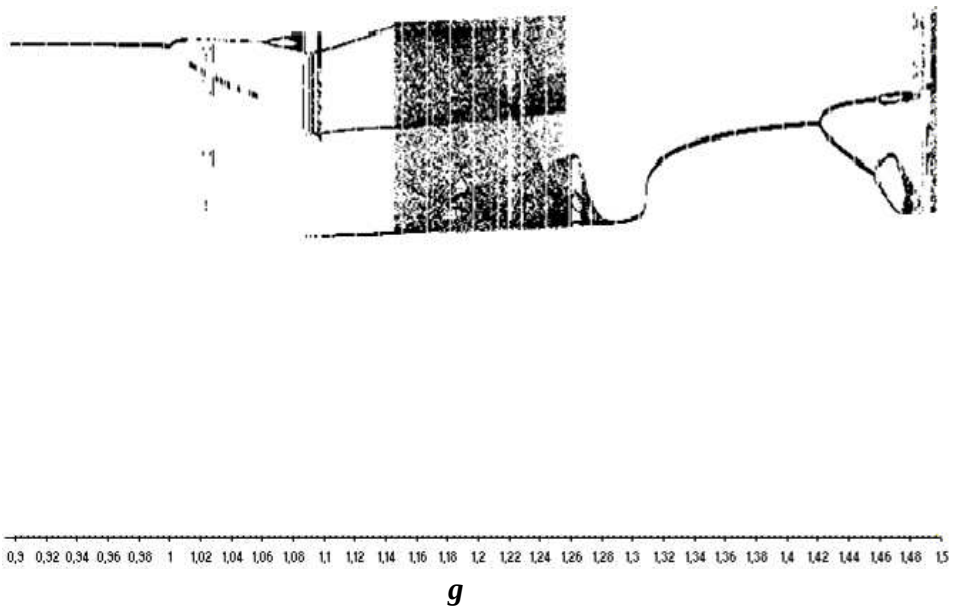
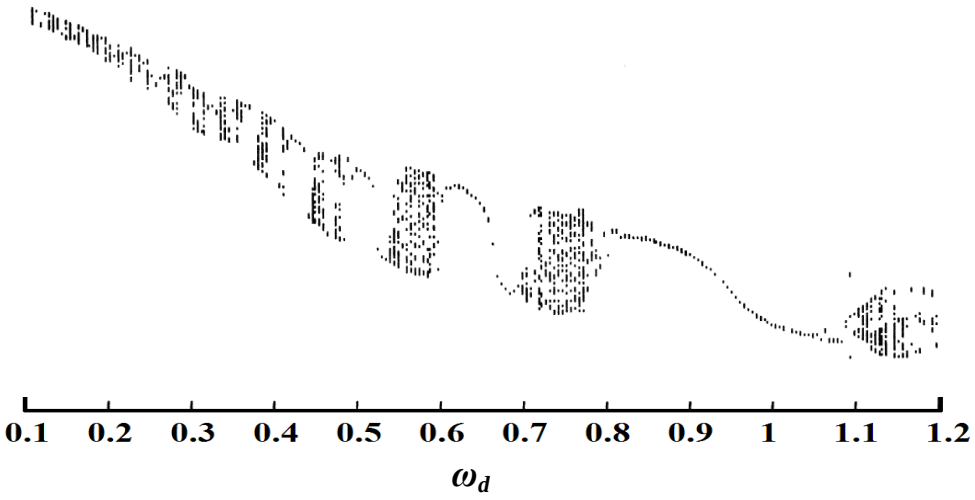


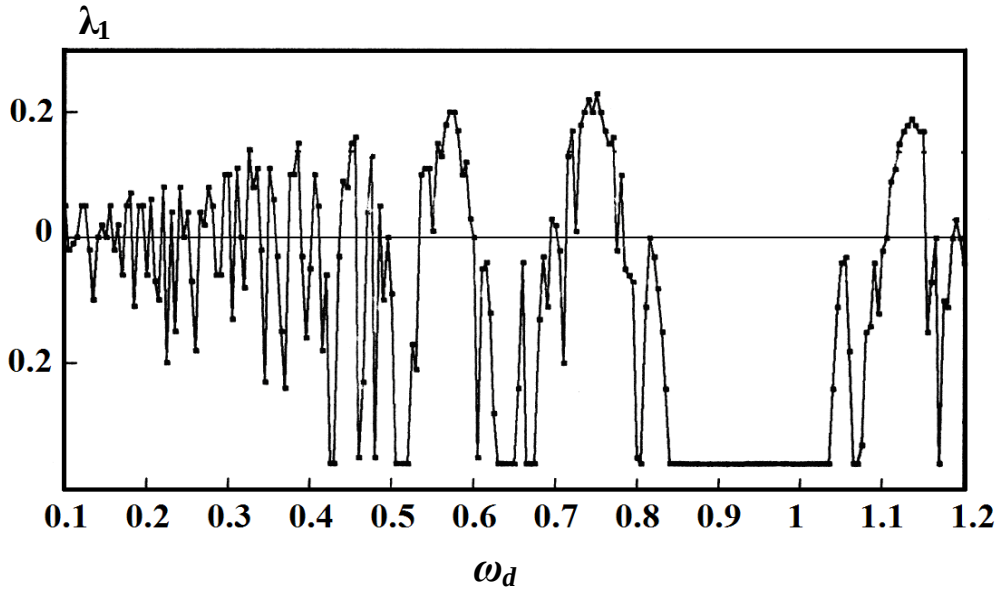
Figure 4.9 Josephson junction system bifurcation diagram vs. microwave amplitude g [$\Delta g = 0.001$]; $k = 0.5$, $\omega_d = 2/3$.

In the second simulation of a Josephson single junction the applied external microwave normalized frequency ω_d is varied from 0.1 to 1.2 with 0.01 increments while the external microwave amplitude and dissipation coefficient are set to 3.8 and 0.5 respectively. Figure 4.10 (a), (b) and (c) demonstrate the bifurcation diagram, the first Lyapunov exponent and LZ complexity in the aforementioned range of parameters. In periodic windows LE is negative and there is a low level of LZ complexity.

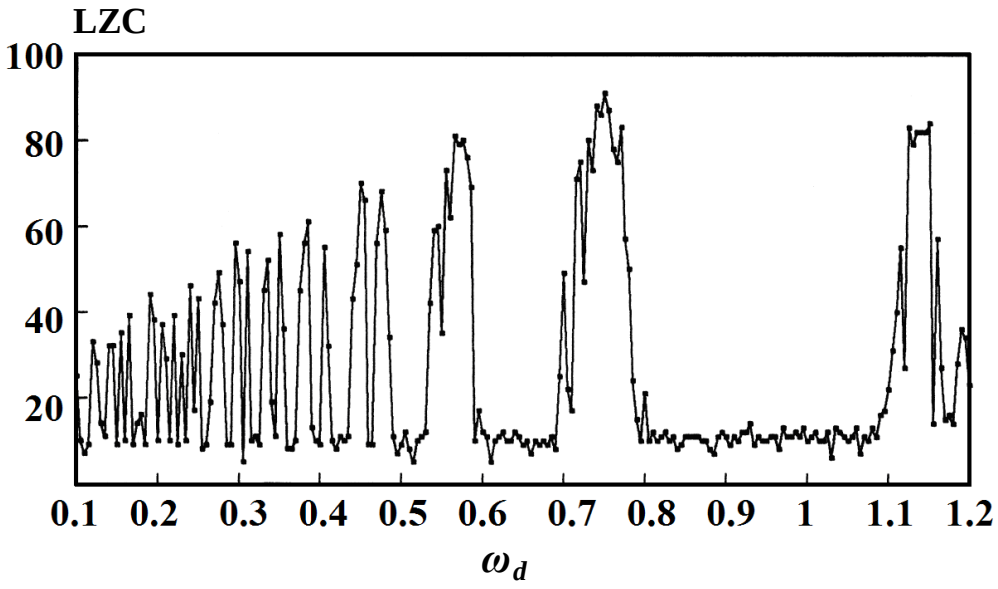


(a)

Figure 4.10 Josephson junction system (a) bifurcation diagram vs. microwave frequency ω_d. Microwave amplitude and system dissipation coefficient are set to 3.8 and 0.5 respectively.



(b)



(c)

Figure 4.10, cont. Josephson junction system (b) first LE (c) LZC vs. microwave frequency ω_d. Microwave amplitude and system dissipation coefficient are set to 3.8 and 0.5 respectively.

4.5 Lorentz System

The Lorenz system is a benchmark in studying chaotic phenomena. This section concentrates on analytical and computer-assisted numerical analysis of this system described by the following differential equations:

$$\frac{dx}{dt} = \sigma(y - x),$$
$$\frac{dy}{dt} = x(\rho - z) - y,$$
$$\frac{dz}{dt} = xy - \beta y.$$

where the parameters σ, β and ρ are real and positive. The Lorenz equations are an example of coupled differential equations with complicated asymptotic behavior. The values initially chosen by Lorenz are $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$. In the context of bifurcation theory it is usual to treat σ and β as fixed and allow ρ to vary. The Lorenz equations are symmetric under the transformation operation from (x, y, z) to $(-x, -y, z)$. Therefore, the phase portrait of the Lorenz system always exhibits this symmetry for all values of ρ. When $\rho < 1$, the only stationary point is at the origin $(0, 0, 0)$ which corresponds to the solutions of $y = x$ (from $\frac{dx}{dt} = 0$), $\beta z = x^2$ (from $\frac{dz}{dt} = 0$) and $\beta(\rho - 1)x - x^3 = 0$, (from $\frac{dy}{dt} = 0$). This corresponds to no convection and the energy is transported only by conduction. There are two other stationary points at $C^+ = \left(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1\right)$ and $C^- = \left(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho - 1\right)$, provided $\rho > 1$. Linear analysis shows that the origin is a stable point if $0 < \rho < 1$ and that the system loses stability in a pitchfork bifurcation at $\rho = 1$, creating the two non-trivial stationary points which are (initially) stable. This is because of the characteristic equation which has three real roots for ρ

> 0. If $\rho < 1$ they are all negative, so the heat conduction is stable. But one root becomes positive if $\rho > 1$, which corresponds to Rayleigh's instability. This implies that the condition for the onset of convection in the Lorenz system is the same as the Rayleigh's result for the full system of hydrodynamic equations. To determine the stability of these stationary points we look at the Jacobian matrix:

$$J = \begin{pmatrix} -\sigma & \sigma & 0 \\ (\rho - z) & -1 & -x \\ y & x & -\beta \end{pmatrix}.$$

which has eigenvalues given by the roots of the characteristic equation, $s^3 + (\sigma + \beta + 1)s^2 + \beta(\rho + \sigma)s + 2\sigma\beta(\rho - 1) = 0$, when evaluated at either stationary point at C^+ and C^-. Since the stationary points are the image of each other under the symmetry, the stability properties of the two stationary points must be the same. Next, we determine the bifurcation of the stationary points, i.e., values of the parameters for which either $s = 0$ or $s = i\omega$ are solutions of the eigenvalue equation. Setting $s = 0$ we find that $\rho = 1$, giving the pitchfork bifurcation. Setting $s = i\omega$ and equating real and imaginary parts of the equation we find:

$$-(\sigma + \beta + 1)\omega^2 + 2\sigma\beta(\rho - 1) = 0, -\omega^3 + \beta(\rho + \sigma)\omega = 0.$$

or

$$\omega^2 = \frac{2\sigma\beta(\rho - 1)}{(\sigma + \beta + 1)} = \beta(\rho + \sigma)\omega.$$

This implies that there is a Hopf bifurcation at

$$\rho_{Hopf} = \frac{\sigma(\sigma + \beta + 3)}{(\sigma - \beta - 1)}.$$

when $\omega^2 > 0$ at this value of ρ, which means if $\omega^2 = \frac{2\beta\sigma(\sigma + 1)}{(\sigma - \beta - 1)} > 0$.

Since ρ_{Hopf} must be greater than one, this stability can occur only if $\sigma > \beta + 1$ is satisfied. For the standard choice of parameter values, $\sigma = 10$ and $\beta = 8/3$ this is clearly satisfied with the value of $\rho_{\text{Hopf}} \approx 22.74$. This is the critical value of ρ for the instability of steady convection. It is possible to show that the Hopf bifurcation is subcritical, creating an unstable periodic orbit in $\rho < \rho_{\text{Hopf}}$. For $\rho > \rho_{\text{Hopf}}$ both stationary points are saddles and so there are no stable periodic orbits and no stable stationary points. This presents us with two problems: it is easy to show that for $\rho < 1$ the origin is globally asymptotically stable and so there are no periodic orbits. But the Hopf bifurcation implies that a pair of periodic orbits which are images of each other under the transformation (x, y, z) to $(-x,-y, z)$ bifurcate for $\rho < \rho_{\text{Hopf}}$, so they must be destroyed in some bifurcation below $\rho = 1$. Also, a bounding function can be constructed for all $\rho > 0$ and so for $\rho > \rho_{\text{Hopf}}$ there are no solutions remaining bounded and yet no stable or attracting objects. These two observations will be resolved by looking at the effect of a homoclinic bifurcation at the origin for some value ρ_{h} in $(1, \rho_{\text{Hopf}})$. Numerical evidence for the existence of a homoclinic bifurcation is obtained by computing the unstable manifold of the origin. Notice that a homoclinic orbit is an orbit $\varphi(x, t)$ such that $\varphi(x, t) \to x_0$ as $t \to \infty$ and $t \to -\infty$, where x_0 is a stationary point of the flow $(x \neq x_0)$. Figures 4.11 (a) and (b) show the positive branch of the unstable manifold at two parameter values, $\rho = 13.8$ and $\rho = 14$. At $\rho = 13.8$ the manifold spirals to the stationary point C^+ (which is stable) without crossing into $x < 0$, while at $\rho = 14$ the unstable manifold spirals into C^- after crossing into the region with $x < 0$. By continuity of solutions with parameters we conclude that for some value between 13.8 and 14, that is, $\rho = \rho_h$ there is a homoclinic orbit biasymptotic to the origin and further that due to the symmetry of the equations there is a second homoclinic orbit involving the negative branch of the unstable manifold of the origin at the same parameter

value, giving the configuration shown in Figure 4.11 (c). The Jacobian matrix at the origin is given by:

$$J = \begin{pmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{pmatrix}.$$

The eigenvectors of Lorenz flow satisfy:

$$\begin{pmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = s \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

The characteristic eigenvectors at the origin are (0, 1, 1), (σ, $\sigma + s_+$, 0) and (σ, $\sigma + s_-$, 0), so that there is one eigenvalue of $s = -\beta$, with the z-axis as its associated eigenvector and the other eigenvalues that are the solutions of

$$s^2 + (1+\sigma)s - \sigma(\rho - 1) = 0,$$

$$s_\pm = \left(\frac{1}{2}\right)\left(-(\sigma+1) \pm \sqrt{(\sigma+1)^2 + 4\sigma(\rho-1)}\right),$$

which gives $s_- < 0$ and $s_+ > 0$. Indeed, for the parameter values considered here $s_- < -\beta < 0 < s_+$, and thus the dominant eigenvalues are $-\beta$ and s_+, and the homoclinic orbit approaches the origin tangential to the z-axis as $t \to \infty$. In this case numerical experiments suggest that both homoclinic orbits approach the origin tangential to the z-axis from above. Finally we note that the saddle index, $\Delta = \beta / s_+$, is less than 1. This information is enough for us to start investigating the effect of a homoclinic bifurcation to the origin involving two homoclinic orbits.

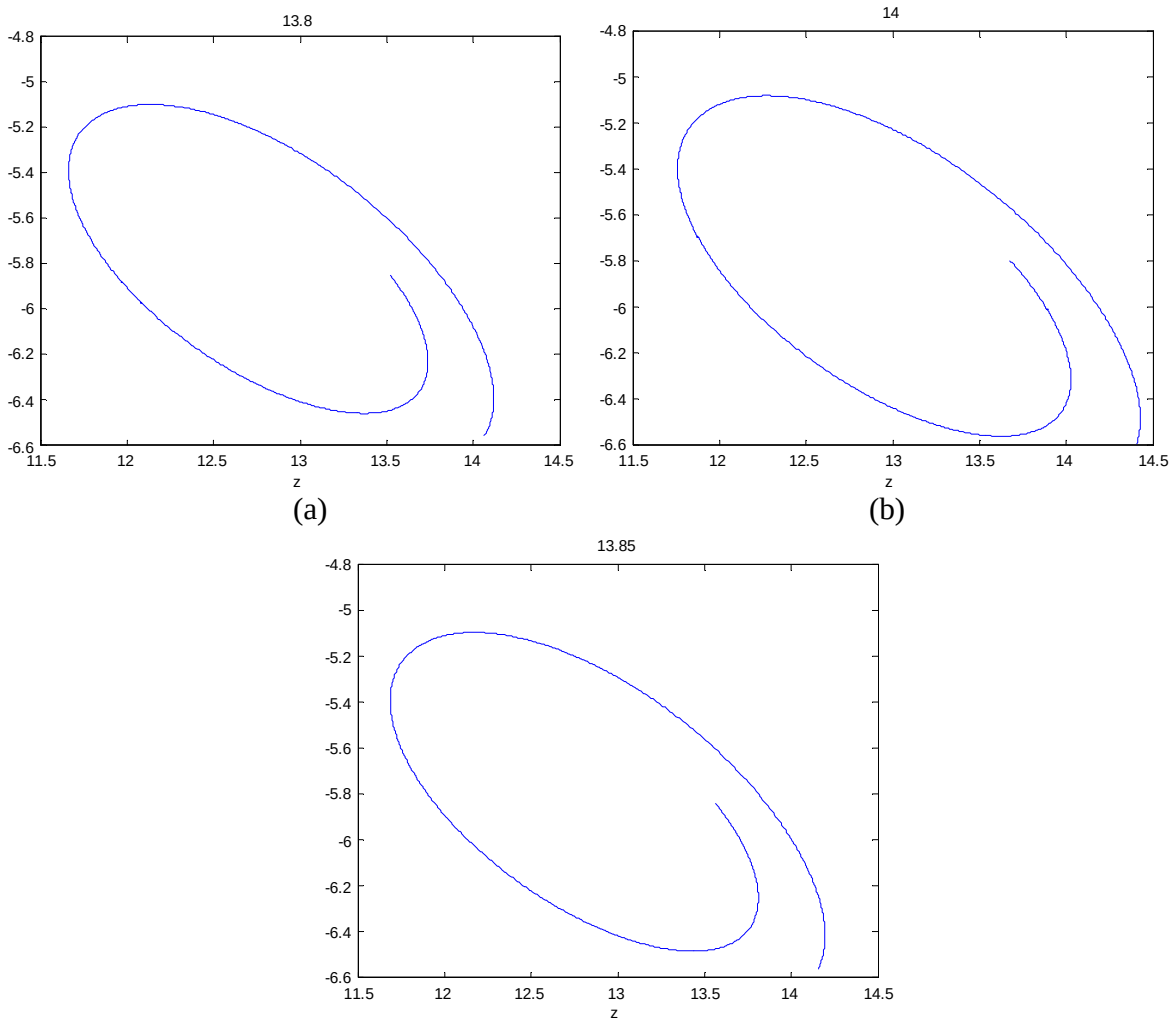


Figure 4.11 One branch of the unstable manifold for (a) $\rho < \rho_h$ (b) $\rho > \rho_h$ and (c) $\rho = \rho_h$.

In order to determine the variation of both the stable and unstable eigenvectors in the (x, y) plane, as ρ varies from 1 to 100, we write the tangent of the angle with the x-axis

$$\tan\Theta = 1 + \left(\frac{s}{\sigma}\right) = \left(\frac{1}{2\sigma}\right)\left((\sigma - 1) \pm \sqrt{(1-\sigma)^2 + 4\sigma\rho}\right),$$

$$\tan\Theta\big|_{\sigma=10} = \left(\frac{9}{20}\right)\left(1 \pm \sqrt{1 + \frac{40\rho}{81}}\right),$$

when $\sigma = 10$. The variation is shown in Figure 4.12 vs. ρ.

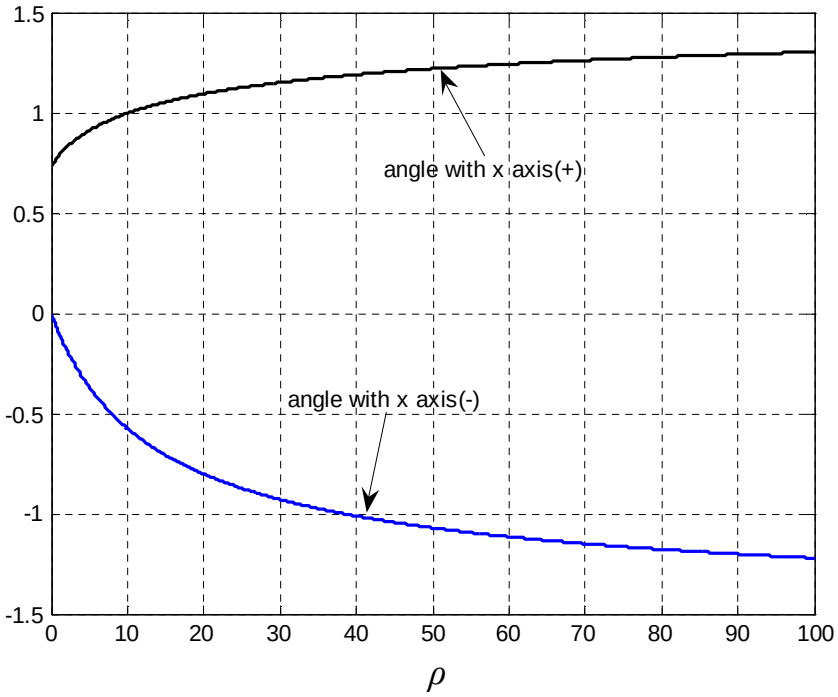


Figure 4.12 Variation of the angle with the Lorenz system x-axis in the $0 \leq \rho \leq 100$ range.

In Figure 4.13 the motion of the trajectories for parameters $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$ is shown. It has been projected onto the (z, y) and (x, y) planes with initial condition (10, −10, 10) over 1000 iterations with 0.01 integration time step and the results are in Figure 4.14. The points (C^+, C^-) are the fixed points: ($6\sqrt{2}$, $6\sqrt{2}$, 27) and ($-6\sqrt{2}$, $-6\sqrt{2}$, 27). The dynamics takes the system in a growing spiral away from one fixed point and drops into the vicinity of the other fixed point and again proceeds to spiral away from it. This flip-flop behavior continues as the dynamic proceeds. This means vertical cylinders change their rotation rate in an increasingly oscillatory manner in time, and then completely reverse their sense of rotation and settle to a relatively stationary state.

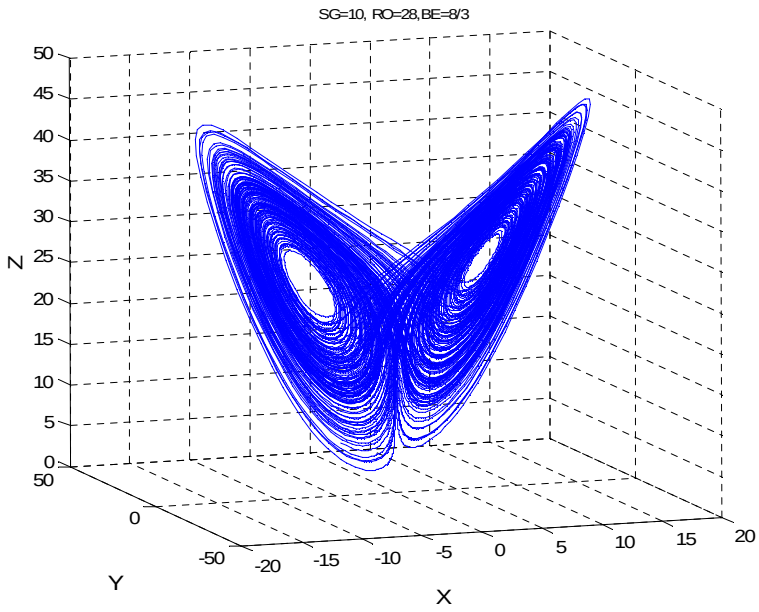


Fig. 4.13 Lorenz attractor for $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$.

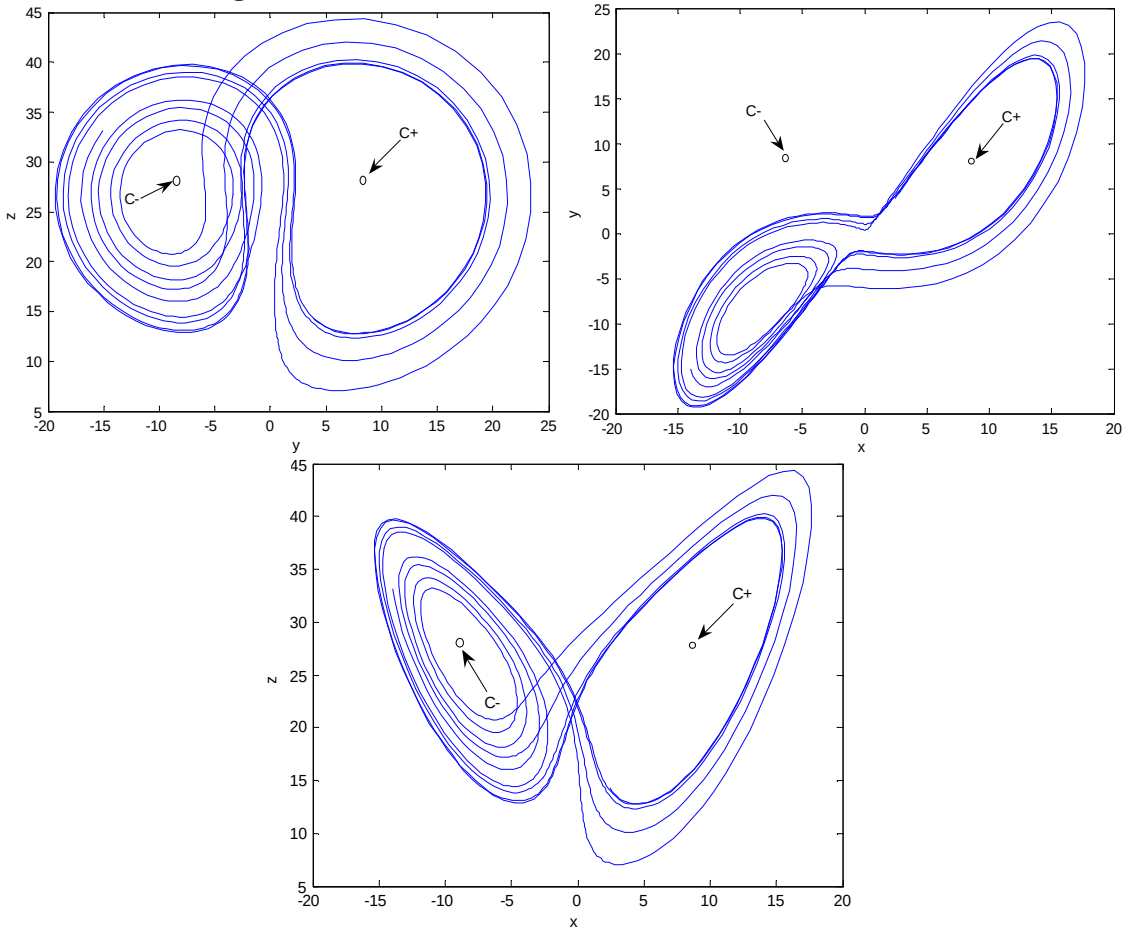


Figure 4.14 Lorenz attractor projection for $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$ (a) z-y plane (b) y-x plane (c) z-x plane.

To illustrate the application of LE estimation with Gram-Schmidt reorthonormalization (GSR) approach and the group theory method with no rescaling and no orthogonalization (NRNO) for a system with more than two degrees of freedom, we turn to the standard test case of the Lorenz equations. Consider the dynamics when ρ varies from zero to 500. This range includes the global bifurcation which occurs for $\rho < \rho_{\text{Hopf}}$, and chaotic dynamics. For this three degree of freedom system the equations are six partially coupled differential equations. In this case

$$\Omega = \Psi^{(1,2)}\Psi^{(1,3)}\Psi^{(2,3)} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{pmatrix},$$

with Jacobian matrix:

$$J = \begin{pmatrix} -\sigma & \sigma & 0 \\ (\rho - z) & -1 & -x \\ y & x & -\beta \end{pmatrix}.$$

The system of equations governing the evolution of the three Lyapunov exponents $\lambda_1, \lambda_2, \lambda_3$ and three angles α, β, γ are:

$$\begin{aligned} \frac{d\lambda_1}{dt} &= \Delta_{11}, \\ \frac{d\lambda_2}{dt} &= \Delta_{22}, \\ \frac{d\lambda_3}{dt} &= \Delta_{33}, \\ \frac{d\alpha}{dt} &= \omega_1(\dot{\alpha}) = \Delta_{21}, \\ \frac{d\beta}{dt} &= \omega_2(\dot{\beta}) = \Delta_{31}, \\ \frac{d\gamma}{dt} &= \omega_3(\dot{\gamma}) = \Delta_{32}. \end{aligned} \tag{4.1}$$

The control parameters, are set to σ = 10 and β = 8/3 while ρ is varied over the range $0 \le \rho \le 500$ with step $\Delta\rho$ = 0.5 and averaged over 10^6 iterations. 2×10^4 initial solutions are discarded for each parameter value. In the Lorenz system of equations, the sum of the three Lyapunov exponents must equal $-(\sigma+\beta+1)$; this is because from Liouville theorem, all volume elements contract in the Lorenz flow. That is $\nabla.\vec{V} = \frac{\partial}{\partial x}(\dot{x}) + \frac{\partial}{\partial y}(\dot{y}) + \frac{\partial}{\partial z}(\dot{z}) = -\sigma - 1 - \beta < 0$, which is independent of *x, y, z*. Therefore, the contraction is uniform in phase space. This means the volume into which the trajectories tend must be zero. The Lyapunov exponents sum vs. GSR inner loop integration steps is plotted and compared to the divergence of dynamical flow. An extensive comparison of the NRNO method against the standard GSR method has been carried out. In both NRNO and GSR methods the sum of the three Lyapunov exponents $-(\sigma+\beta+1) = -13.67$ at σ = 10, β = 8/3 is maintained. Numerically we confirm that the best convergence occurs when the GSR inner loop integration number is one **(**Figure 4.15). Furthermore, in order to determine the best integration time step in Lyapunov spectrum calculation, we extend the analysis to include the variation of outer loop integration time step. Both methods were applied to the same trajectories generated using a fourth order Runge-Kutta integrator applied to the Lorenz system equations with time step independent of the sampling interval over the investigated range $0.001 \le t_s \le 0.02$. Error and convergence analysis has been carried out by applying the two methods to the trajectories sampled over time intervals $t_s \ge h$. Both methods has been implemented with h = 0.001 and both generate essentially identical results. Figure 4.17 (a) and (b) illustrates the Lorenz system LE spectrum and LEs sum versus ρ when it changes from 0 to 500 with increment 0.5. For this 3-dimensional system, execution times for both methods are similar.

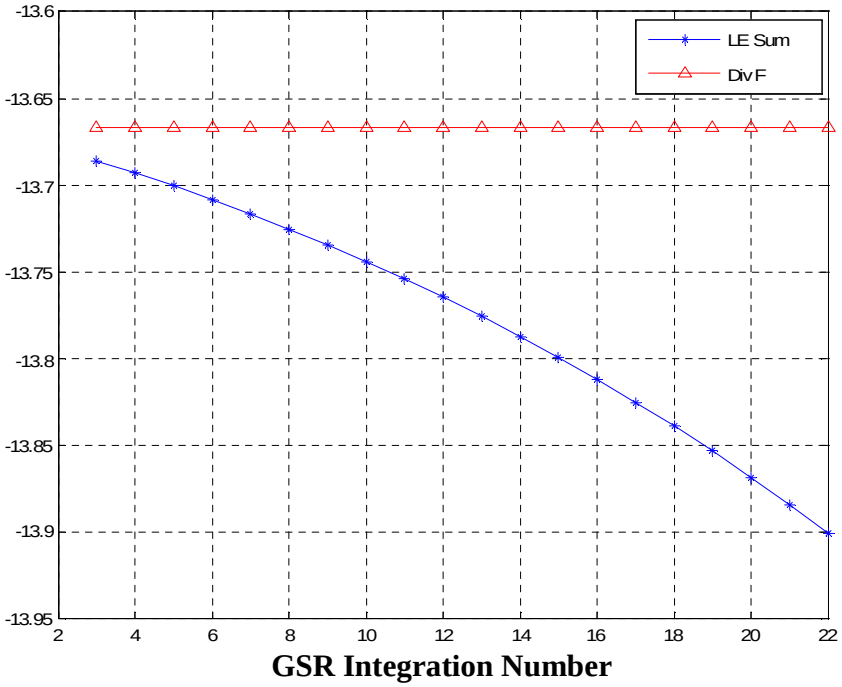


Figure 4.15 Comparision of Lorenz system LE sum and flow divergence vs. GSR integration number for fixed parameters value, $\sigma = 10$, $\beta = 8/3$, $\rho = 25$.

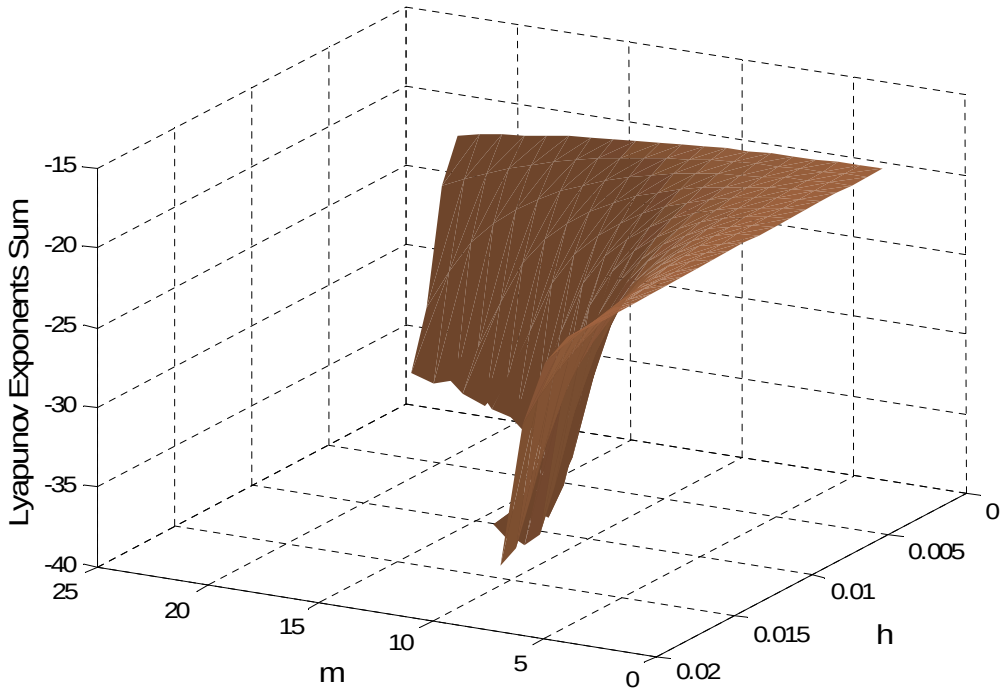
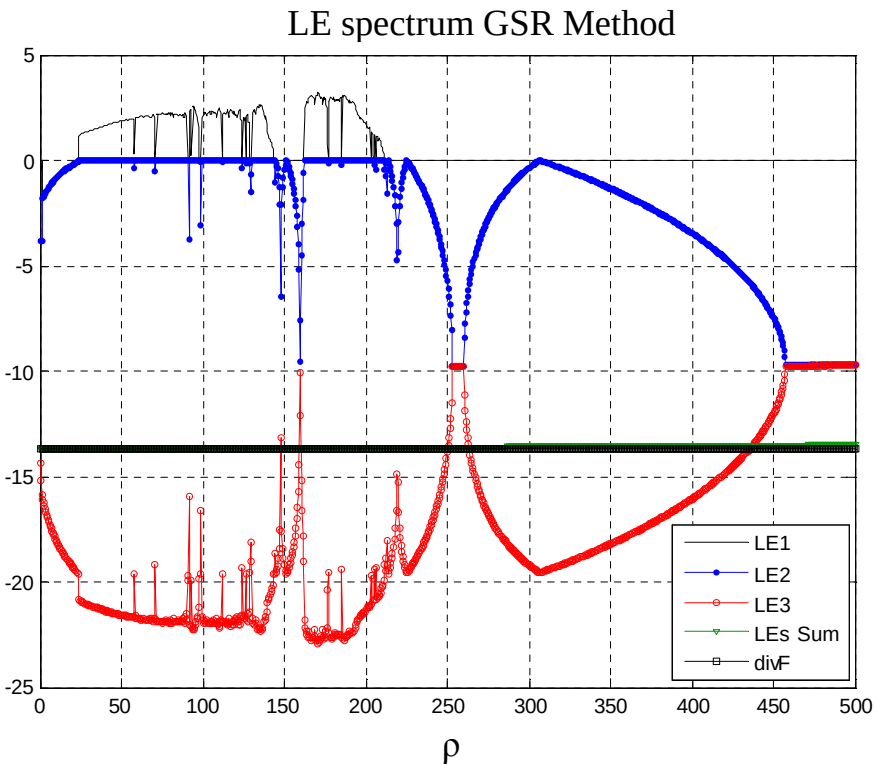
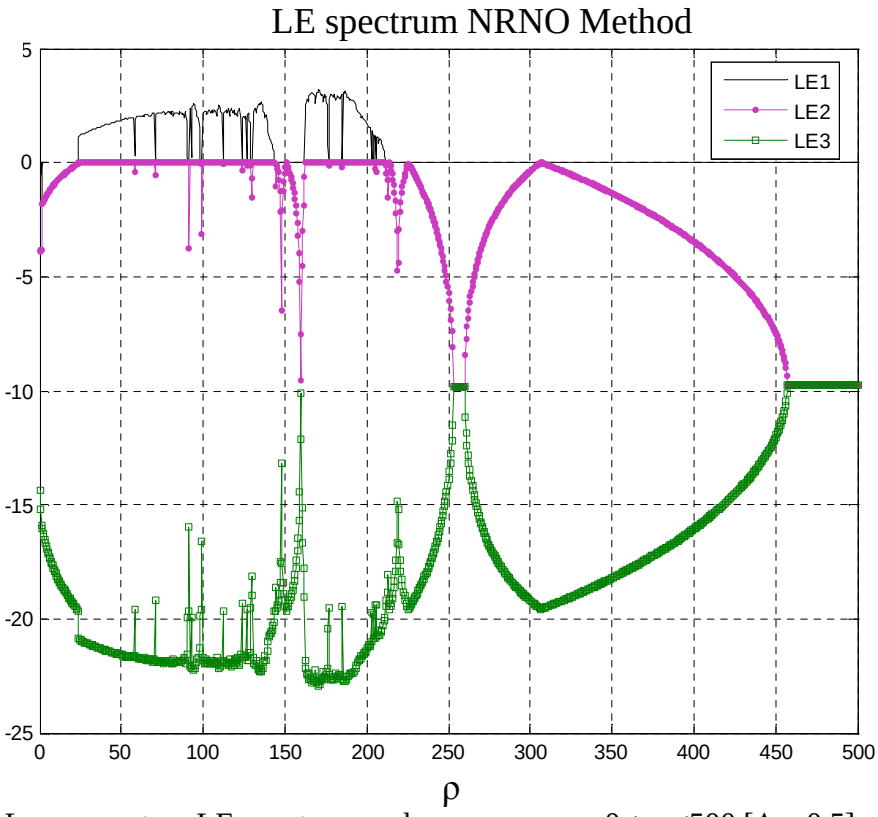


Figure 4.16 Comparision of Lorenz system LE sum and flow divergence vs. GSR integration number and dynamic system integration time step. $\sigma = 10$, $\beta = 8/3$, $\rho = 25$.



(a)



(b)

Figure 4.17 Lorenz system LE spectrum and sum versus ρ , $0< \rho <500$ [$\Delta\rho$=0.5], σ = 10, β=8/3 (a) GSR method (b) NRNO method.

The LZ complexity of the Lorenz system is also calculated in the same parameter range for ρ, when other parameters are set to $\sigma = 10$ and $\beta = 8/3$. ρ is varied each time with a 0.1 increment. Each string includes 32700 phase space bit samples. To collect these samples, phase space is partitioned into eight subspaces. According to following rule, the dynamics of Lorenz system is mapped to the corresponding j^{th} character, s_j, in the buffer sequence, $S = s_1 s_2 \cdots s_j s_{j+1} \cdots s_n$. Bits are generated based on their trajectory location in the phase space grid. Dynamic states move along the unstable direction and loop immediately to the neighborhood of C^-, spiral outwards, then flip over to the region of C^+, and continue to spiral and flip out again. Thus, the partitioning borders are defined as the $z = \rho - 1$, $x = 0$ and $y = 0$ planes. Mapping is done based on the trajectory asymptotic values for a certain control parameter. For σ =10, $\beta = 8/3$ and $\rho = 28$, the grid defines eight regions in phase space for the following decision rule: If the dynamic state belongs to one of following grid zones, $x > 0$, $y > 0$, $z > 27$; $x > 0$, $y < 0$, $z < 27$; $x < 0$, $y > 0$, $z < 27$; $x < 0$, $y < 0$, $z > 27$, insert $s_i = 0$; Otherwise, for $x > 0$, $y > 0$, $z < 27$; $x > 0$, $y < 0$, $z > 27$; $x < 0$, $y > 0$, $z > 27$; $x < 0$, $y < 0$, $z < 27$, insert $s_j = 1$. The result is illustrated in Figure 4.18 and compared to largest Lyapunov exponent in Figure 4.19. The Lorenz bifurcation diagram which is the plot of the maximum value of y for the same parameter values used for computing LZC and LE is shown in Figure 4.20 (a). To remove the unwanted trajectories we choose a filtering window. The trajectories are selected if they satisfy the following decision rule:

Set T = period = 1, time = (integration step size) (iteration number), T_{pp} = time / T

If T_{pp} – INTEGER (T_{pp}) ≤ tolerance, then Select y.

With this algorithm, those trajectories that are confined to the region: $z \geq 0$, $x \geq 0$ and $x \leq$ window size are admitted. In numerical analysis the parameters are set to $N = 1600$, transient

states = 4000, tolerance = 0.01, integration time step = 0.01, $\sigma = 10$, $\beta = 8/3$ and ρ is varied from 0 to 500 with step 0.1. Figure 4.20 (b) shows the bifurcation diagram resulting from application of filtering algorithm.

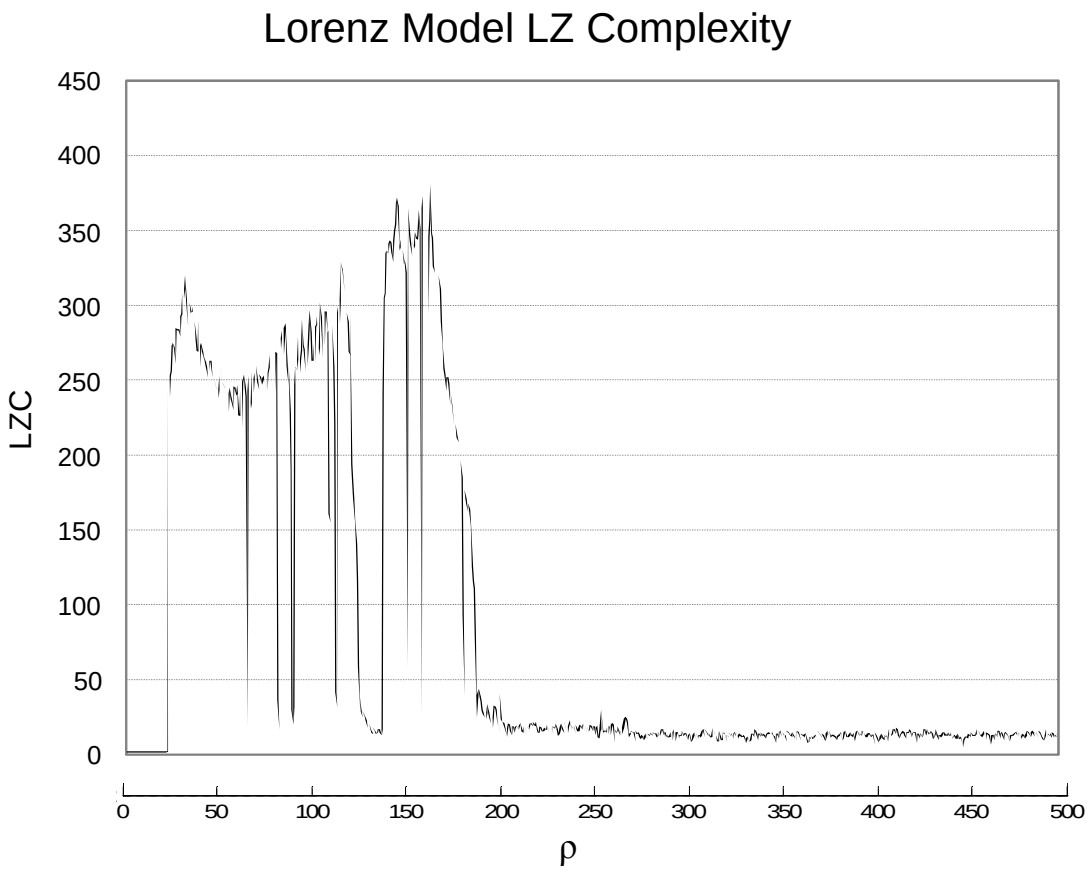


Figure 4.18 Lorenz system LZC. Control parameters $0 \leq \rho \leq 500$ [$\Delta\rho$=0.1], $\sigma = 10$, $\beta = 8/3$.

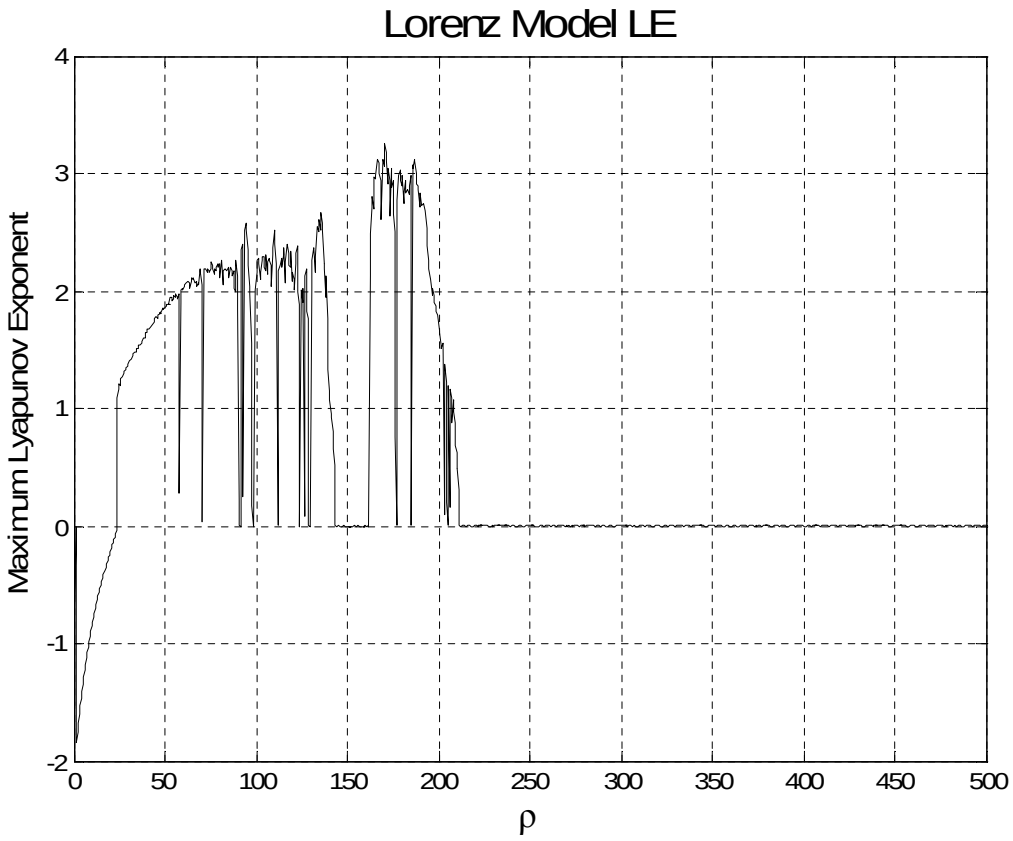


Figure 4.19 Lorenz system first LE. Control parameters $0 \leq \rho \leq 500$ [$\Delta\rho$=0.1], $\sigma = 10$, $\beta = 8/3$.

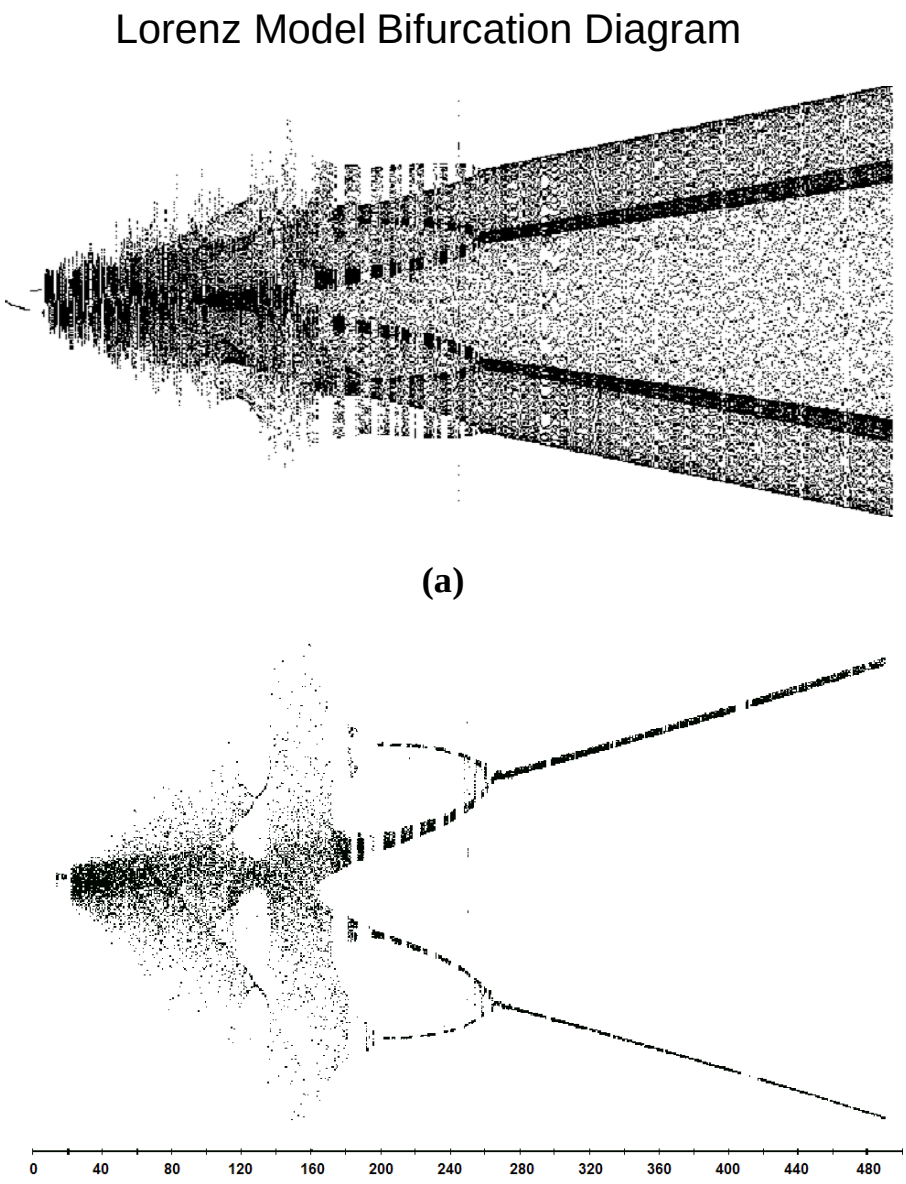


Figure 4.20 Lorenz bifurcation diagram. (a) without window (b) with window, size = 5. Control parameters: $0 \leq \rho \leq 500$ [$\Delta\rho$=0.1]; σ = 10, β = 8/3, N = 1600, transient states = 4000; tolerance=0.01, integration time step= 0.01.

CHAPTER 5

ULTRA HIGH-SPEED MICROBRIDGE CHAOS DOMAIN

5.1 Introduction

In this chapter we study the dynamical behavior of the Josephson tetrode microbridge system [89]. The model device consists of five Josephson weak-link junctions connected through microbridge configurations with four-terminals. Two junctions are connected in series and three junctions are connected in parallel. The first Lyapunov exponent is calculated through Gram-Schmidt reorthonormalization (GSR) algorithm from the dynamics of electrical voltages across the junctions when the normal resistance between terminal 1 and 4 is varied. Simulation results confirm the existence of chaotic, quasiperiodic and periodic oscillations in this device. The resulting high frequency chaotic oscillations are proposed for applications such as ultra high frequency pseudorandom code-generation in spread-spectrum communications and secret key generation for secure communication channels [90, 91, 92]. In our analysis we have used a resistively and capacitively shunted Josephson junction (RCSJ) model for numerical calculation [93, 94]. Typical superconductors have a coherence length of the order of hundreds of nanometers, which is much greater than the thickness of the usual oxide layers. Therefore, weak links have relatively small specific capacitance *C/A*. Thus, capacitance effect is considered small in a microbridge Josephson junction. This means the resistively shunted Josephson junction (RSJ) model is accurate enough to describe the dynamics of this junction device model. This system consists of five Josephson junctions with three independent variables to satisfy the

necessary condition for chaotic behavior. We numerically analyze the dynamics of the Josephson microbridge to study the nonlinear behavior leading to chaos generation.

5.2 Josephson Microbridge Tetrode System Model

The Josephson tetrode equivalent circuit consists of five Josephson junctions, as shown in Figure 5.1. Superconductive terminals (1, 2, 3, and 4) are weakly connected through five Josephson junctions. Two of the junctions are series-connected (1–2 and 2–3) and the other three junctions are parallel connected (1–4, 2–4, and 3–4) in the Josephson tetrode. In microbridge configurations for the Josephson junctions the capacitance between the microbridges is negligible. Using a conventional RSJ model to describe the dynamics of the Josephson tetrode, three independent differential equations for the phase differences φ_{12} ,φ_{14}, φ_{23} can be written. The other phase variances can be described as dependent phase variances because by our assumption, there is no magnetic flux in the loop of the Josephson junctions, whose length is much smaller than the coherent length. Therefore, the sum of the phase differences in one loop is zero. The equations of the phase difference in terms of normalized explicit functions in the Josephson tetrode and its coefficients are given below. φ_{ij} is the phase difference of the junction, $R_{n,ij}$ is the resistance of the Josephson junction and $r_{nij}=R_{nij}/R_{n12}$ is the normalized resistance. $I_{n,ij}$ is the critical current in the five Josephson junctions, and I_i is the dc drive current. All the critical currents and the dc drive currents are normalized by I_{c12} that is $i_{cij}=I_{cij}/I_{c12}$. We assume Nb as a superconducting material. Thus, the product of all junctions is constant near the critical temperature for microbridge junctions, i.e., $I_{c,ij}\ R_{n,ij\ =}\ V_{c,ij}$ is a constant value for microbridge junctions. We set $V_{c,ij}$ to 0.45 mV. Then the critical currents are $i_{c,ij}\, r_{n,ij} = 1$, $i_1=I_1/I_{c12}$; $i_2=I_2/I_{c12}$

and $i_3=I_3/I_{c12}$. Then for microbridge junctions we have $i_{c12}=1/r_{n12}$; $i_{c23}=1/r_{n23}$; $i_{c24}=1/r_{n24}$; $i_{c34}=1/r_{n34}$.

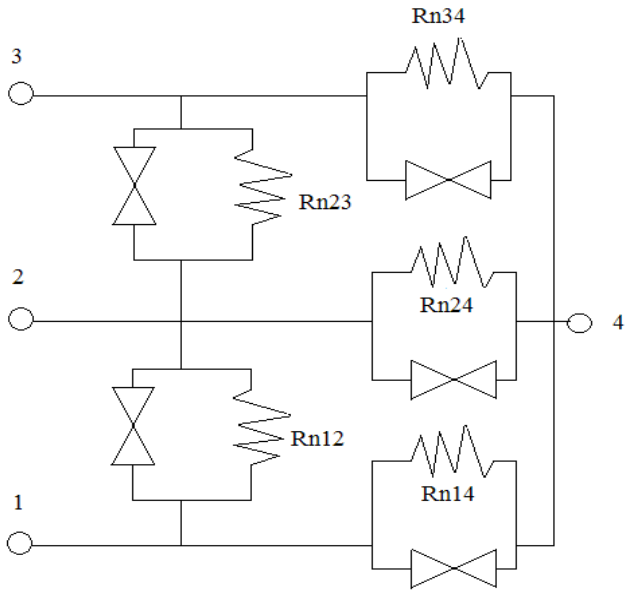


Figure 5.1 Equivalent circuit configuration of the Josephson Tetrode Microbridge device.

The differential equations which model the system using the resistively and capacitively shunted Josephson junction (RCSJ) model are

$$\left(\frac{\hbar C_{14}}{2e}\right)\frac{\mathrm{d}^2\varphi_{14}}{\mathrm{d}t^2}+\left(\frac{\hbar}{2eR_{n14}}\right)\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}t}+I_{c14}\sin\varphi_{14}+\left(\frac{\hbar C_{12}}{2e}\right)\frac{\mathrm{d}^2\varphi_{12}}{\mathrm{d}t^2}+\left(\frac{\hbar}{2eR_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}+$$
$$I_{c12}\sin\varphi_{12}-\left(\frac{\hbar C_{12}}{2e}\right)\frac{\mathrm{d}^2\varphi_{12}}{\mathrm{d}t^2}-\left(\frac{\hbar}{2eR_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}-I_{c12}\sin\varphi_{12}=I_1(t),$$

$$\left(\frac{\hbar C_{23}}{2e}\right)\frac{\mathrm{d}^2\varphi_{23}}{\mathrm{d}t^2}+\left(\frac{\hbar}{2eR_{n23}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}+I_{c23}\sin\varphi_{23}+\left(\frac{\hbar C_{24}}{2e}\right)\frac{\mathrm{d}^2\varphi_{24}}{\mathrm{d}t^2}+\left(\frac{\hbar}{2eR_{n24}}\right)\frac{\mathrm{d}\varphi_{24}}{\mathrm{d}t}$$
$$+I_{c24}\sin\varphi_{24}-\left(\frac{\hbar C_{12}}{2e}\right)\frac{\mathrm{d}^2\varphi_{12}}{\mathrm{d}t^2}-\left(\frac{\hbar}{2eR_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}-I_{c12}\sin\varphi_{12}=I_2(t),$$

$$\left(\frac{\hbar C_{34}}{2e}\right)\frac{\mathrm{d}^2\varphi_{34}}{\mathrm{d}t^2}+\left(\frac{\hbar}{2eR_{n34}}\right)\frac{\mathrm{d}\varphi_{34}}{\mathrm{d}t}+I_{c34}\sin\varphi_{34}-\left(\frac{\hbar C_{23}}{2e}\right)\frac{\mathrm{d}^2\varphi_{23}}{\mathrm{d}t^2}$$
$$-\left(\frac{\hbar}{2eR_{n23}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}-I_{c23}\sin\varphi_{23}=I_3(t).$$

(5.1)

Microbridge junctions may exhibit an inductive part which may lead to a phase shift between current and voltage. In this analysis we ignore the small inductive effects due to a substantial contribution to the junction inductance from the kinetic inductance. Also we assume that the capacitance between the microbridges is negligible. Thus, the conventional RSJ model is used to describe the dynamics of the Josephson tetrode. Note that Φ_0 is flux quantum.

$$\left(\frac{1}{R_{n14}C_{14}}\right)\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}t}+\left(\frac{2\pi I_{c14}}{\Phi_0 C_{14}}\right)\sin\varphi_{14}+\left(\frac{1}{R_{n12}C_{14}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}+\left(\frac{2\pi I_{c12}}{\Phi_0 C_{14}}\right)\sin\varphi_{12}=\left(\frac{2\pi}{\Phi_0 C_{14}}\right)I_1(t),$$

$$\left(\frac{1}{R_{n14}C_{23}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}+\left(\frac{2\pi I_{c23}}{\Phi_0 C_{14}}\right)\sin\varphi_{23}+\left(\frac{1}{R_{n24}C_{23}}\right)\frac{\mathrm{d}\varphi_{24}}{\mathrm{d}t}+\left(\frac{2\pi I_{c24}}{\Phi_0 C_{23}}\right)\sin\varphi_{24}-\left(\frac{1}{R_{n12}C_{23}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}$$
$$-\left(\frac{2\pi I_{c12}}{\Phi_0 C_{23}}\right)\sin\varphi_{12}=\left(\frac{2\pi}{\Phi_0 C_{23}}\right)I_2(t), \qquad (5.2)$$

$$\left(\frac{1}{R_{n34}C_{34}}\right)\frac{\mathrm{d}\varphi_{34}}{\mathrm{d}t}+\left(\frac{2\pi I_{c34}}{\Phi_0 C_{34}}\right)\sin\varphi_{34}-\left(\frac{1}{R_{n23}C_{34}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}-\left(\frac{2\pi I_{c23}}{\Phi_0 C_{34}}\right)\sin\varphi_{23}=\left(\frac{2\pi}{\Phi_0 C_{34}}\right)I_3(t).$$

Assuming that all junctions have identical capacitance, we write:

$$\left(\frac{1}{R_{n14}}\right)\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}t}+\left(\frac{2\pi I_{c14}}{\Phi_0}\right)\sin\varphi_{14}+\left(\frac{1}{R_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}+\left(\frac{2\pi I_{c12}}{\Phi_0}\right)\sin\varphi_{12}=\left(\frac{2\pi}{\Phi_0}\right)I_1(t),$$

$$\left(\frac{1}{R_{n14}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}+\left(\frac{2\pi I_{c23}}{\Phi_0}\right)\sin\varphi_{23}+\left(\frac{1}{R_{n24}}\right)\frac{\mathrm{d}\varphi_{24}}{\mathrm{d}t}+\left(\frac{2\pi I_{c24}}{\Phi_0}\right)\sin\varphi_{24}-\left(\frac{1}{R_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}$$
$$-\left(\frac{2\pi I_{c12}}{\Phi_0}\right)\sin\varphi_{12}=\left(\frac{2\pi}{\Phi_0}\right)I_2(t),$$

$$\left(\frac{1}{R_{n34}}\right)\frac{\mathrm{d}\varphi_{34}}{\mathrm{d}t}+\left(\frac{2\pi I_{c34}}{\Phi_0}\right)\sin\varphi_{34}-\left(\frac{1}{R_{n23}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}-\left(\frac{2\pi I_{c23}}{\Phi_0}\right)\sin\varphi_{23}=\left(\frac{2\pi}{\Phi_0}\right)I_3(t).$$

(5.3)

Multiplying by $\Phi_0/2\pi$, we obtain

$$\left(\frac{\Phi_0}{2\pi R_{n14}}\right)\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}t}+I_{c14}\sin\varphi_{14}+\left(\frac{\Phi_0}{2\pi R_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}+I_{c12}\sin\varphi_{12}=I_1(t),$$

$$\left(\frac{\Phi_0}{2\pi R_{n14}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}+I_{c23}\sin\varphi_{23}+\left(\frac{\Phi_0}{2\pi R_{n14}}\right)\frac{\mathrm{d}\varphi_{24}}{\mathrm{d}t}+I_{c24}\sin\varphi_{24}-\left(\frac{\Phi_0}{2\pi R_{n12}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}$$
$$-I_{c12}\sin\varphi_{12}=I_2(t), \qquad (5.4)$$

$$\left(\frac{\Phi_0}{2\pi R_{n34}}\right)\frac{\mathrm{d}\varphi_{34}}{\mathrm{d}t}+I_{c34}\sin\varphi_{34}-\left(\frac{\Phi_0}{2\pi R_{n23}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}-I_{c23}\sin\varphi_{23}=I_3(t).$$

Using the aforementioned approximation and substituting $\varphi_{24}=\varphi_{14}-\varphi_{12}$ and $\varphi_{34}=\varphi_{14}-\varphi_{12}-\varphi_{23}$ in (Eqs. 5.4), the system of equations can be written as:

$$\eta\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}\tau}=\alpha_{12}+\beta_{12}\sin\varphi_{12}+\gamma_{12}\sin\varphi_{23}+\delta_{12}\sin\varphi_{14}+\sigma_{12}\sin(\varphi_{14}-\varphi_{12})+\rho_{12}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23}),$$
$$\eta\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}\tau}=\alpha_{23}+\beta_{23}\sin\varphi_{12}+\gamma_{23}\sin\varphi_{23}+\delta_{23}\sin\varphi_{14}+\sigma_{23}\sin(\varphi_{14}-\varphi_{12})+\rho_{23}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23}),$$
$$\eta\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}\tau}=\alpha_{14}+\beta_{14}\sin\varphi_{12}+\gamma_{14}\sin\varphi_{23}+\delta_{14}\sin\varphi_{14}+\sigma_{14}\sin(\varphi_{14}-\varphi_{12})+\rho_{14}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23}), \qquad (5.5)$$

where $\eta = h\omega_J/4\pi eI_{c12}R_{n12}$ and $\omega_J = (2\pi I_{c12}/\Phi_0 C_{12})^{1/2}$ is the junction plasma frequency. Also we define $\tau = \omega_J t$. In addition by using the normalized critical currents $i_{cij}=I_{cij}/I_{c12}$ and normalized resistances $r_{nij}=R_{nij}/R_{n12}$, the following parametric coefficients are defined.

$$\zeta=\left\{\left(r_{n34}+r_{n23}\right)\left(1+\frac{1+r_{n14}}{r_{n24}}+r_{n14}+1\right)\right\}^{-1},$$
$$\alpha_{12}=-\zeta\left(r_{n14}+r_{n14}\left(\frac{r_{n34}+r_{n23}}{r_{n24}}\right)i_1-\zeta\left(r_{n34}+r_{n23}\right)i_2-\zeta r_{n34}i_3\right),$$
$$\beta_{12}=-\zeta\left(r_{n14}+r_{n14}\left(\frac{r_{n34}+r_{n23}}{r_{n24}}\right)-\left(r_{n34}+r_{n23}\right)\right),$$
$$\gamma_{12}=\rho_{12}=\zeta,$$

$$\delta_{12} = -\zeta\left(1+\frac{r_{n34}+r_{n23}}{r_{n24}}\right),$$

$$\sigma_{12} = \zeta\left(\frac{r_{n34}+r_{n23}}{r_{n24}}\right),$$

$$\alpha_{23} = \left(\zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)\left(r_{n14}+r_{n14}\frac{r_{n34}+r_{n23}}{r_{n24}}\right)-\frac{r_{n14}r_{n23}}{r_{n24}}\right)i_1 +$$

$$+\left(r_{n23}-\zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)\left(r_{n34}+r_{n23}\right)\right)i_2-\zeta r_{n23}r_{n34}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)i_3,$$

$$\beta_{23} = -\zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)\left(r_{n14}+r_{n34}+r_{n23}+r_{n14}\left(\frac{r_{n34}+r_{n23}}{r_{n24}}\right)\right)+\frac{r_{n14}r_{n23}}{r_{n24}}+r_{n23},$$

$$\gamma_{23} = \zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)-1,$$

$$\delta_{23} = -\zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)\left(1+\frac{r_{n34}+r_{n23}}{r_{n24}}\right)+\frac{r_{n23}}{r_{n24}},$$

$$\sigma_{23} = \zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right)\left(\frac{r_{n34}+r_{n23}}{r_{n24}}\right)-\frac{r_{n23}}{r_{n24}},$$

$$\rho_{23} = \zeta r_{n23}\left(1+\frac{1+r_{n14}}{r_{n24}}\right),$$

(5.6)

$$\alpha_{14} = \left(r_{n14}-\zeta r_{n14}\left(r_{n14}+\frac{r_{n14}\left(r_{n34}+r_{n23}\right)}{r_{n24}}\right)\right)i_1+\zeta r_{n14}\left(r_{n34}+r_{n23}\right)i_2+\zeta r_{n14}r_{n34}i_3,$$

$$\beta_{14} = -r_{n14}+\zeta r_{n14}\left(r_{n14}+r_{n34}+r_{n23}+\frac{r_{n14}\left(r_{n34}+r_{n23}\right)}{r_{n24}}\right),$$

$$\gamma_{14} = -\zeta r_{n14},$$

$$\delta_{14} = \zeta r_{n14}\left(1+\frac{r_{n34}+r_{n23}}{r_{n24}}\right)-1,$$

$$\sigma_{14} = -\zeta r_{n14}\left(\frac{r_{n34}+r_{n23}}{r_{n24}}\right),$$

$$\rho_{14} = -\zeta r_{n14}.$$

(5.7)

5.3 Microbridge Tetrode Model Simulation Results

We integrate the dynamic equations to determine the temporal waveforms of the normalized voltages. The junction resistances are $R_{n,12} = R_{n,23} = R_{n,24} = 10\ \Omega$. The dc bias currents set to $I_1 = I_2 = I_3 = 1.1I_{c12}$. Figure 5.2 shows temporal waveforms of the three normalized voltages at different $R_{n,14}$. Chaotic behavior may be observed from the irregularity of the output values. The waveforms become more chaotic as $R_{n,14}$ is decreased from 1.4. The temporal waveform fluctuates chaotically in the range 0.4 to 1.5 Ω. As it is decreased further from 0.4, the temporal waveforms become quasi-periodic oscillations again, and a stable output is obtained. When $R_{n,14}$ is increased above 1.4 quasi-periodic oscillations return. In the region above $R_{n,14} = 2.8$, behavior is completely cyclic. The oscillation frequencies of the chaotic temporal waveforms are around hundreds of gigahertz, which could be useful for cryptosystem key generation by generating ultra fast random-like signal.

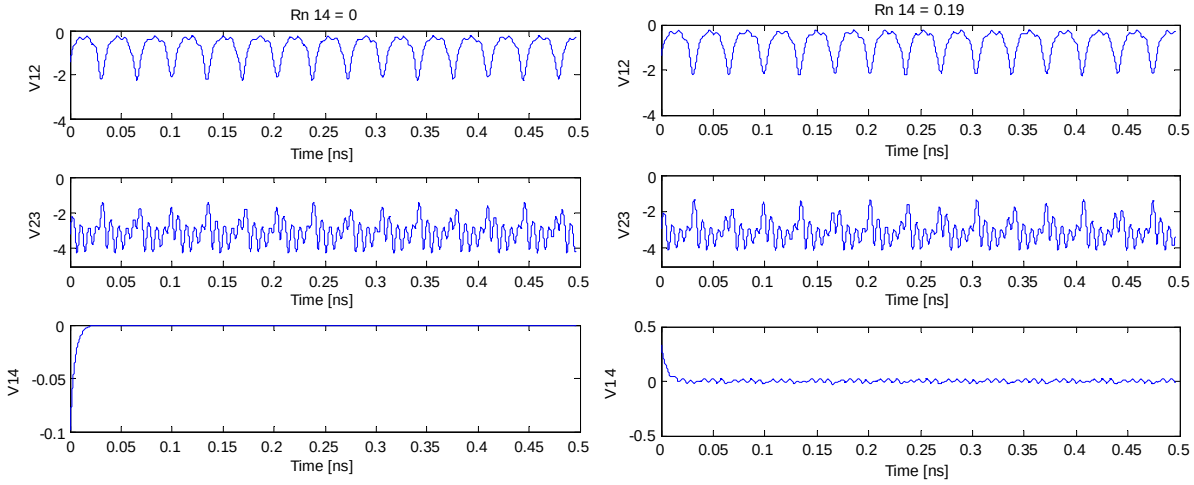


Figure 5.2 Waveforms of the tetrode microbridge electrical voltages at various $R_{n,14}$. Quasi-periodic oscillations at $R_{n,14} = 0$, $R_{n,14} = 0.19$, $R_{n,14} = 0.4$, $R_{n,14} = 1.5$, $R_{n,14} = 2.2$ and $R_{n,14} = 2.8$. Chaotic oscillations phase portraits corresponding to $R_{n,14} = 0.7$, $R_{n,14} = 1.25$, $R_{n,14} = 1.4$, and periodic oscillations at $R_{n,14} = 3$.

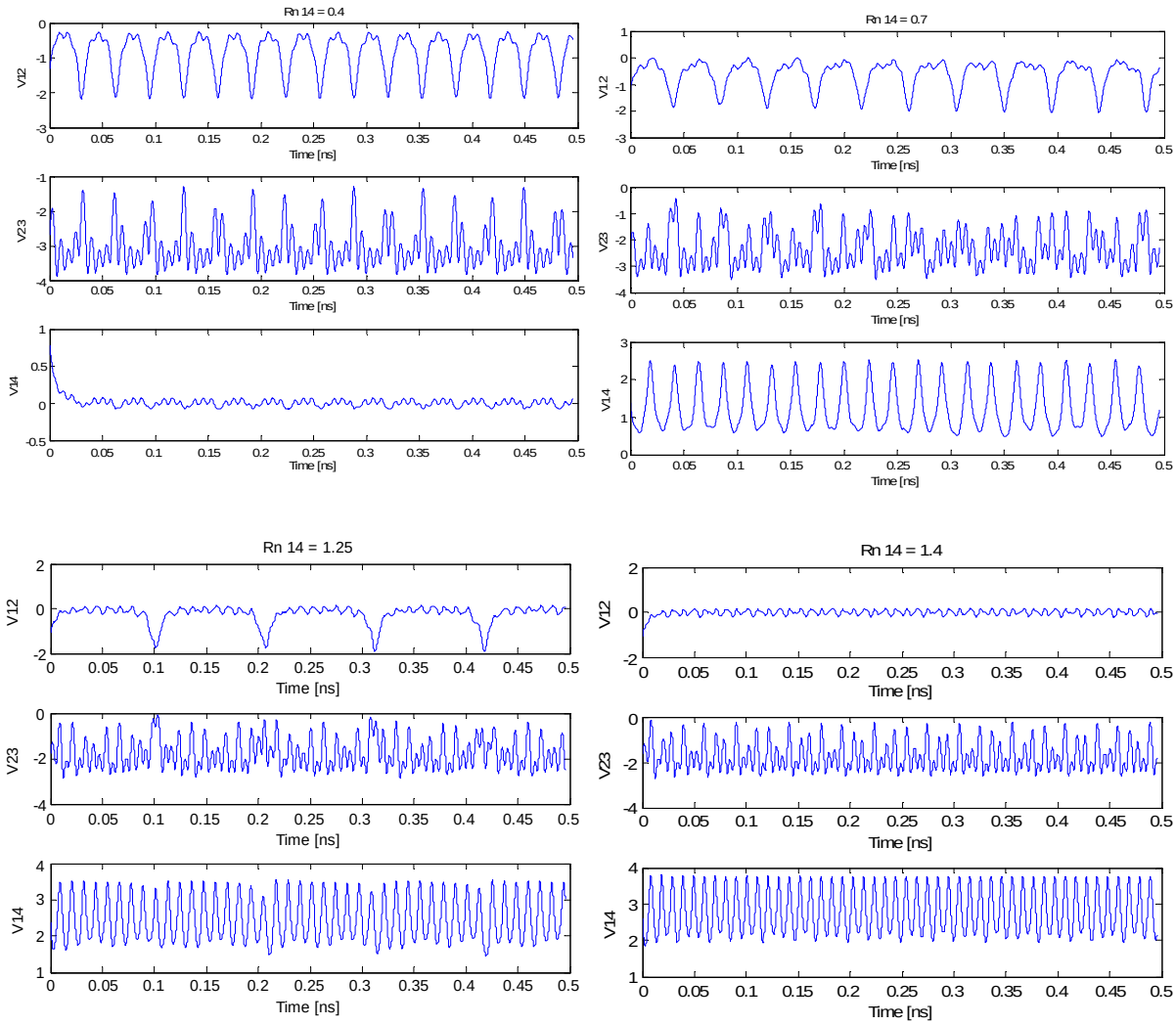


Figure 5.2, cont. Waveforms of the tetrode microbridge electrical voltages at various $R_{n,14}$. Quasi-periodic oscillations at $R_{n,14} = 0$, $R_{n,14} = 0.19$, $R_{n,14} = 0.4$, $R_{n,14} = 1.5$, $R_{n,14} = 2.2$ and $R_{n,14} = 2.8$. Chaotic oscillations phase portraits corresponding to $R_{n,14} = 0.7$, $R_{n,14} = 1.25$, $R_{n,14} = 1.4$, and periodic oscillations at $R_{n,14} = 3$.

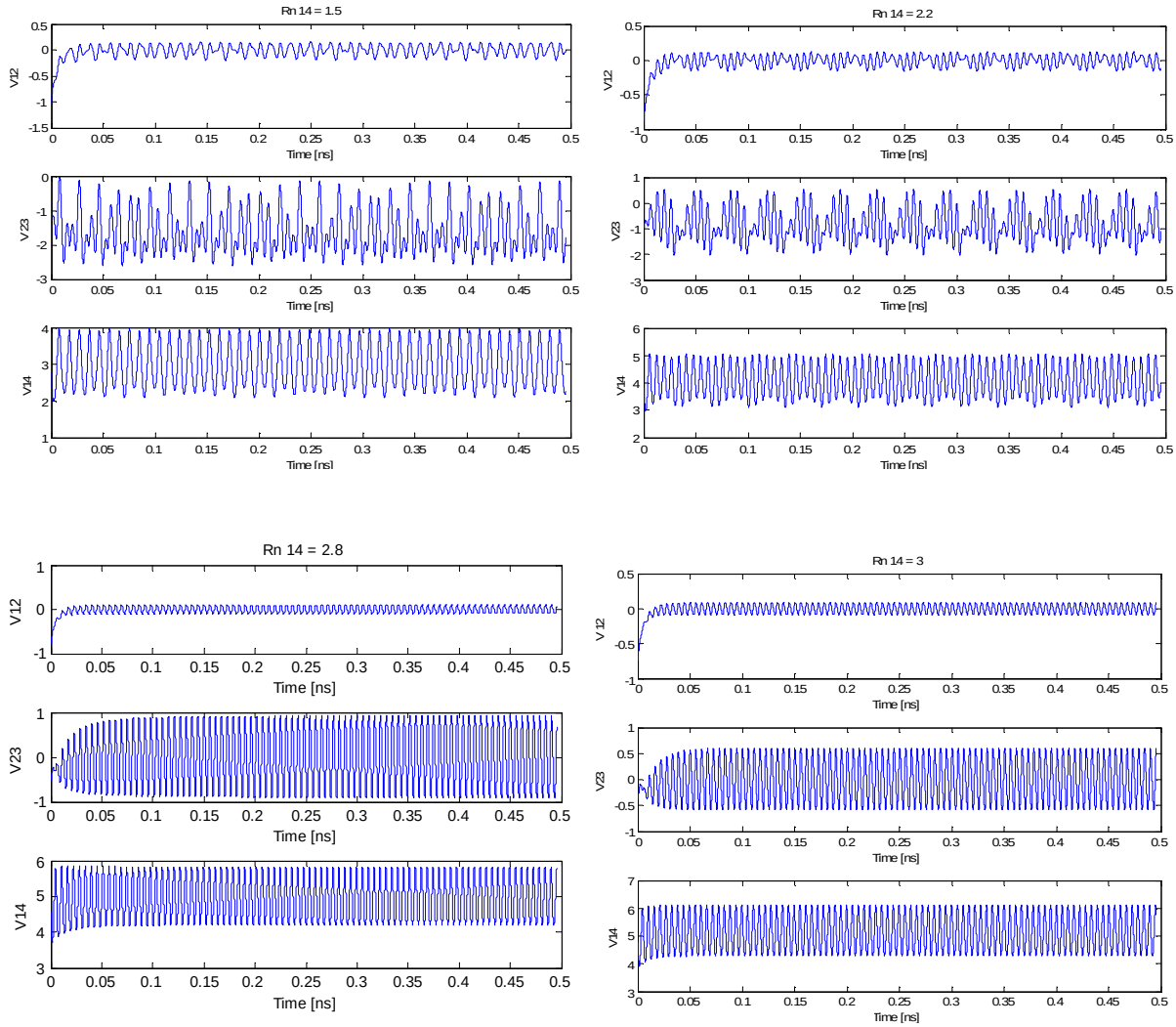


Figure 5.2, cont. Waveforms of the tetrode microbridge electrical voltages at various $R_{n,14}$. Quasi-periodic oscillations at $R_{n,14}$ = 0, $R_{n,14}$ = 0.19, $R_{n,14}$ = 0.4, $R_{n,14}$ = 1.5, $R_{n,14}$ = 2.2 and $R_{n,14}$ = 2.8. Chaotic oscillations phase portraits corresponding to $R_{n,14}$ = 0.7, $R_{n,14}$ = 1.25, $R_{n,14}$ = 1.4, and periodic oscillations at $R_{n,14}$ = 3.

We also calculate the three-dimensional (3-D) attractors in the phase space of the three normalized voltages. Phase portraits corresponding to the temporal waveforms of $V_{ij}(t)$ at different $R_{n,14}$'s is shown in Figure 5.3 for quasi-periodic oscillations at $R_{n,14}$ = 0, 0.19, 0.4, 1.5, 2.2, 2.7 Ω; periodic oscillations at $R_{n,14}$ = 2.8, 3 Ω; and chaotic oscillations corresponding to

$R_{n,14}$ = 0.7, 1.25 and 1.39. As $R_{n,14}$ is decreased from 3, the shape of the quasi-periodic attractor is slightly changed, and it suddenly breaks down at 1.4 Ω.

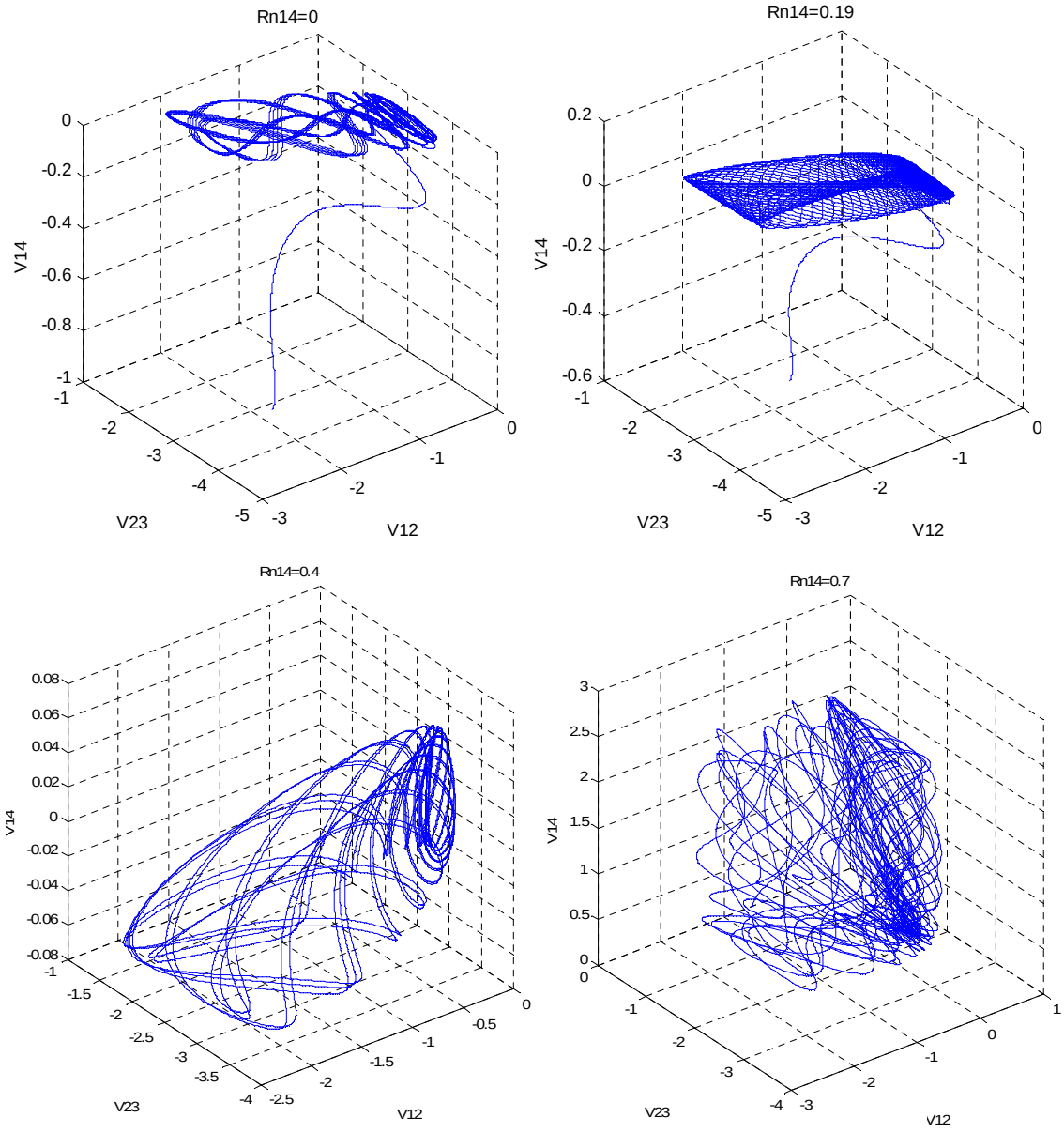


Figure 5.3 Tetrode microbridge electrical voltages phase portraits. Quasi-periodic oscillations at $R_{n,14}$ = 0, 0.19, 0.4, 1.5, 2.2 and 2.8. Chaotic oscillations phase portraits corresponding to $R_{n,14}$ = 0.7, 1.25, 1.4 and periodic oscillation $R_{n,14}$ = 3.

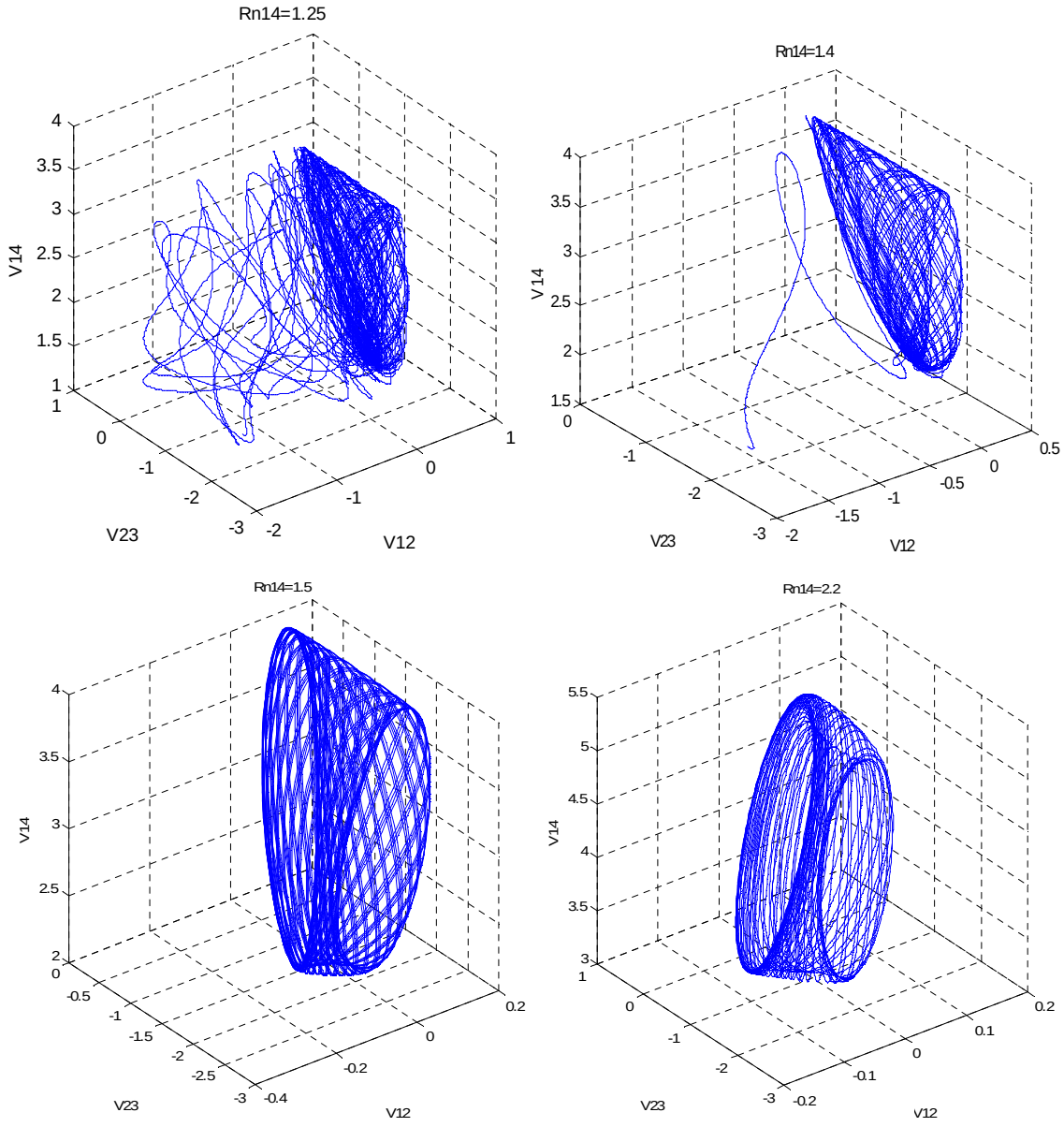


Figure 5.3, cont. Tetrode microbridge electrical voltages phase portraits. Quasi-periodic oscillations at $R_{n,14}$ = 0, 0.19, 0.4, 1.5, 2.2 and 2.8. Chaotic oscillations phase portraits corresponding to $R_{n,14}$ = 0.7, 1.25, 1.4 and periodic oscillation $R_{n,14}$ = 3.

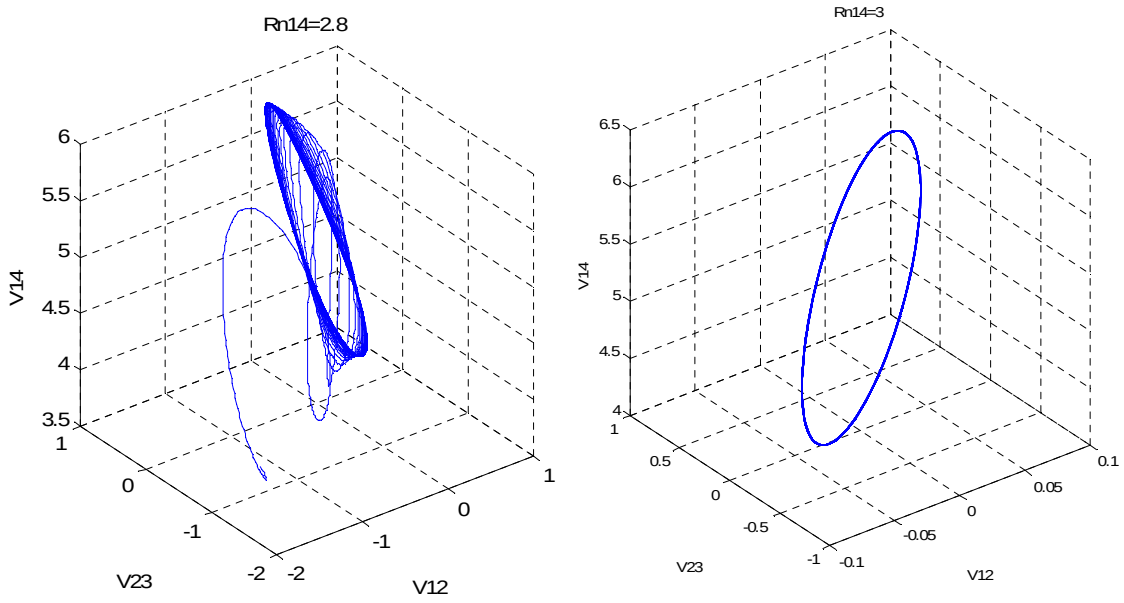


Figure 5.3, cont. Tetrode microbridge electrical voltages phase portraits. Quasi-periodic oscillations at $R_{n,14}$ = 0, 0.19, 0.4, 1.5, 2.2 and 2.8. Chaotic oscillations phase portraits corresponding to $R_{n,14}$ = 0.7, 1.25, 1.4 and periodic oscillation $R_{n,14}$ = 3.

5.4 Tetrode Microbridge Lyapunov Exponent

We have calculated the Lyapunov exponent for a Josephson tetrode versus the control parameter $R_{n,14}$. From equation (5.5) the dynamical flow is

$$\vec{F}(\bar{X},\vec{X}_0,\vec{C}) = \frac{d\vec{X}}{d\tau} = \left(\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}\tau}, \frac{\mathrm{d}\varphi_{23}}{\mathrm{d}\tau}, \frac{\mathrm{d}\varphi_{14}}{\mathrm{d}\tau}\right),$$

$$F_1 = \eta\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}\tau} = \alpha_{12} + \beta_{12}\sin\varphi_{12} + \gamma_{12}\sin\varphi_{23} + \delta_{12}\sin\varphi_{14} + \sigma_{12}\sin(\varphi_{14}-\varphi_{12}) + \rho_{12}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23}),$$

$$F_2 = \eta\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}\tau} = \alpha_{23} + \beta_{23}\sin\varphi_{12} + \gamma_{23}\sin\varphi_{23} + \delta_{23}\sin\varphi_{14} + \sigma_{23}\sin(\varphi_{14}-\varphi_{12}) + \rho_{23}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23}),$$

$$F_3 = \eta\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}\tau} = \alpha_{14} + \beta_{14}\sin\varphi_{12} + \gamma_{14}\sin\varphi_{23} + \delta_{14}\sin\varphi_{14} + \sigma_{14}\sin(\varphi_{14}-\varphi_{12}) + \rho_{14}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23}).$$

(5.8)

And the divergence of the flow can be written as

$$\nabla.\vec{F} = \ln 2 \sum_{i=1}^{\dim \vec{F}=3} \lambda_i = \left(\frac{1}{\eta}\right)\left(\beta_{12}\cos\varphi_{12} - \sigma_{12}\cos(\varphi_{14}-\varphi_{12}) - \rho_{12}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23})\right) +$$
$$\left(\frac{1}{\eta}\right)\left(\gamma_{23}\cos\varphi_{23} - \rho_{23}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right) +$$
$$\left(\frac{1}{\eta}\right)\left(\delta_{14}\cos\varphi_{14} + \sigma_{14}\cos(\varphi_{14}-\varphi_{12}) + \rho_{14}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right). \tag{5.9}$$

Then the elements of the Jacobian matrix are

$$J_{11} = \left(\frac{1}{\eta}\right)\left(\beta_{12}\cos\varphi_{12} - \sigma_{12}\cos(\varphi_{14}-\varphi_{12}) - \rho_{12}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{12} = \left(\frac{1}{\eta}\right)\left(\gamma_{12}\cos\varphi_{23} - \rho_{23}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{13} = \left(\frac{1}{\eta}\right)\left(\delta_{12}\cos\varphi_{14} + \sigma_{12}\cos(\varphi_{14}-\varphi_{12}) + \rho_{12}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{21} = \left(\frac{1}{\eta}\right)\left(\beta_{23}\cos\varphi_{12} - \sigma_{23}\cos(\varphi_{14}-\varphi_{12}) - \rho_{23}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{22} = \left(\frac{1}{\eta}\right)\left(\gamma_{23}\cos\varphi_{23} - \rho_{23}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{23} = \left(\frac{1}{\eta}\right)\left(\delta_{23}\cos\varphi_{14} + \sigma_{23}\cos(\varphi_{14}-\varphi_{12}) + \rho_{23}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{31} = \left(\frac{1}{\eta}\right)\left(\beta_{14}\cos\varphi_{12} - \sigma_{14}\cos(\varphi_{14}-\varphi_{12}) - \rho_{14}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{32} = \left(\frac{1}{\eta}\right)\left(\gamma_{12}\cos\varphi_{23} - \rho_{23}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right),$$
$$J_{33} = \left(\frac{1}{\eta}\right)\left(\delta_{14}\cos\varphi_{14} + \sigma_{14}\cos(\varphi_{14}-\varphi_{12}) + \rho_{14}\cos(\varphi_{14}-\varphi_{12}-\varphi_{23})\right).$$

(5.10)

Stable periodic attractors have only zero and negative λ. For this system there is only one finite positive Lyapunov exponent and only one symbolic Lyapunov spectrum of the form (+, 0, -). Λ^{+} is for those trajectories that are expanding, while λ^{-} is for others that are contracting. In

Figure 5.4 we have plotted the largest Lyapunov exponent versus junction inductance $R_{n,14}$ respectively. This is compared to bifurcation diagram for voltage V_{12} with integration time step 0.01. Here, 3000 transient solutions are discarded and 4000 solutions are used for each $R_{n,14}$ value from 0 to 3. The tolerance is 0.0005, and the selection time period is $T_p=2\pi$ (Figure 5.5). Figure 5.6 shows the Lempel-Ziv complexity in the same range. The decision mechanism is based on four cell partition including $V_{23} > -2$ and $V_{14} > 2$; $V_{23} < -2$ and $V_{14} > 2$; $V_{23} > -2$ and $V_{14} < 2$; $V_{23} < -2$ and $V_{14} < 2$. The results indicate distinct regions corresponding to chaotic and periodic behaviors. Quasiperiodic and periodic responses are distinguishable from chaotic responses through a sharp drop in the Lyapunov exponent value. Observe that, as the resistance decreases, a transition from quasi-periodic oscillation to chaotic oscillation appears. When the system has periodic or quasi periodic states as displayed in the bifurcation window and phase space portraits, the first LE is negative or close to zero. This corresponds to the limit-cycle and quasi-periodic attractor. On the other hand chaotic attractors with some trajectories expanding, while others are contracting, have positive Lyapunov exponent. This is nicely demonstrated by the increasing value of maximum Lyapunov exponent plot from negative to positive and back to negative and close to zero values for quasi-periodic and periodic behaviors. We observe chaotic dynamics between the two different quasi-periodic attractors when $R_{n,14}$ is changed. This quasi-periodic domain is located in the range of 0 to 0.4 Ω for $R_{n,14}$ with close to zero and negative Lyapunov exponents. Chaotic oscillations are obtained when $R_{n,14}$ is in the range from 0.4 to 1.4 Ω, with a maximum value for Lyapunov exponent at 1.4. As $R_{n,14}$ is increased from 1.4 to 2.8 Ω, quasi-periodic oscillations are observed again. Thus, chaotic oscillations are obtained at the transition between two different quasi-periodic attractors. The transition from quasi-periodic to chaotic attractors indicates that the quasi-periodicity-breakdown route to chaos is occurred rather

than period-doubling route. In the same range, there are some tiny regions of periodic windows. By increasing $R_{n,14}$ from 2.8 Ω we observe many periodic oscillations; one example is $R_{n,14}$ = 3 Ω.

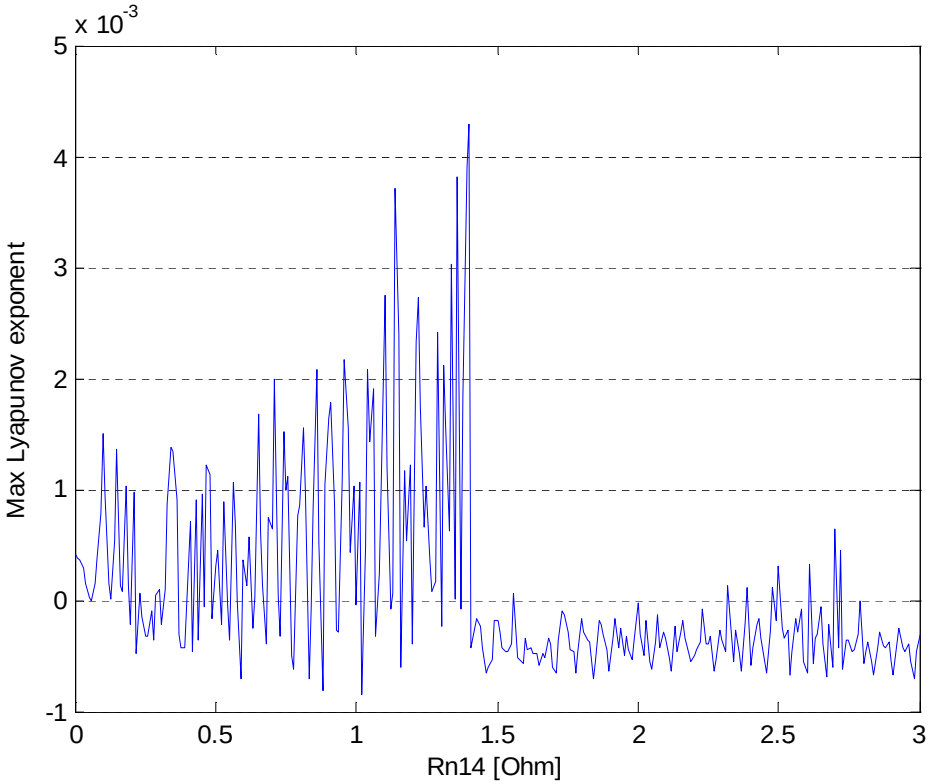


Figure 5.4 Tetrode microbridge LE vs. normal resistance $R_{n,14}$ [$\Delta R_{n,14}$ = 0.01].

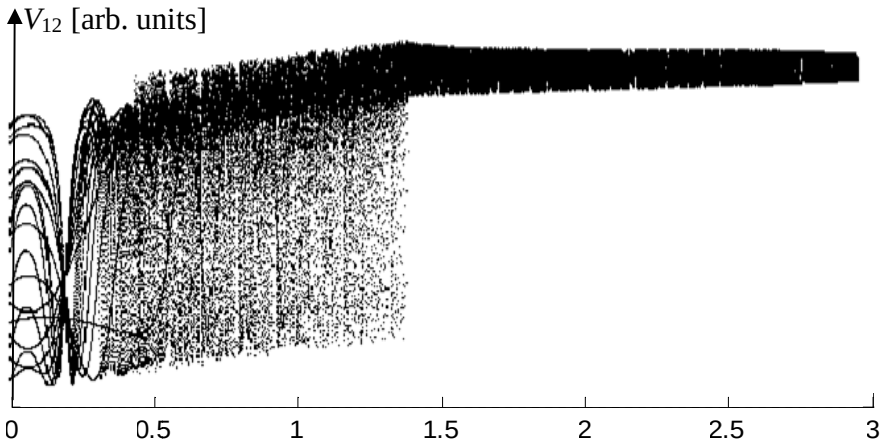


Figure 5.5 Tetrode microbridge bifurcation diagram for voltage V_{12} vs. $R_{n,14}$ with integration time step, 0.01, transients discarded, 3000, integration numbers, 4000, tolerance, 0.0005, T_p=2π.

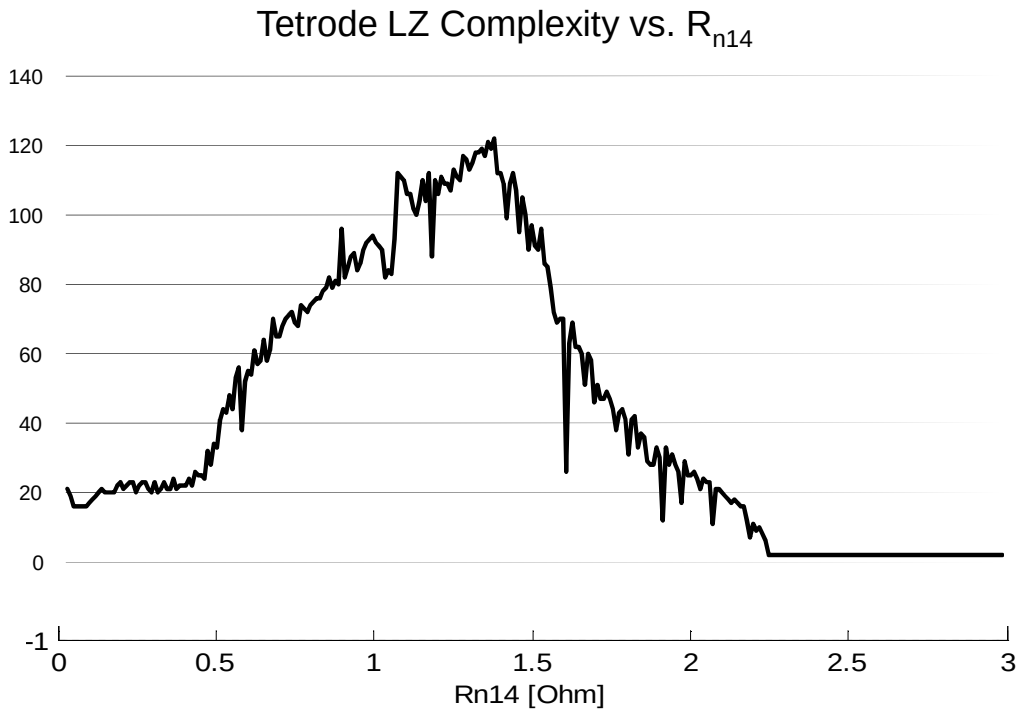


Figure 5.6 Tetrode microbridge LZC plot for voltage V_{23} and V_{14} vs. $R_{n,14}$

Now we investigate the mechanism of the generation of chaos in the Josephson tetrode. We estimate the dominant oscillation frequencies of the three independent voltages. We analytically calculate the oscillation frequencies by using the coefficients of the normalized equations in (Eqs. 5.6). We simplify the normalized equations by neglecting the cross-coupling terms of the other variables as follows:

$$\begin{aligned}
\eta \frac{\mathrm{d}\varphi_{12}}{\mathrm{d}\tau} &= \alpha_{12} + \beta_{12} \sin \varphi_{12}, \\
\eta \frac{\mathrm{d}\varphi_{23}}{\mathrm{d}\tau} &= \alpha_{23} + \gamma_{23} \sin \varphi_{23}, \\
\eta \frac{\mathrm{d}\varphi_{14}}{\mathrm{d}\tau} &= \alpha_{14} + \delta_{14} \sin \varphi_{14}.
\end{aligned} \tag{5.11}$$

From these equations, we analytically estimate the oscillation frequencies of the voltages as follows

$$f_{12} = f\left[(\alpha_{12}/\beta_{12})^2 - 1\right]^{1/2},$$
$$f_{23} = f\left[(\alpha_{23}/\gamma_{23})^2 - 1\right]^{1/2},$$
$$f_{12} = f\left[(\alpha_{14}/\delta_{12})^2 - 1\right]^{1/2}.$$

(5.12)

The oscillation frequencies are non-zero if conditions $\alpha_{12} > \beta_{12}$, $\alpha_{23} > \gamma_{23}$, $\alpha_{14} > \delta_{12}$ are satisfied, otherwise, the oscillation frequencies are zero. This shows that the oscillation frequencies of the voltages are predictable from the coefficients of (5.11). These oscillation frequencies are shown in Figure 5.7. The region for chaotic oscillations in Figure 5.4 and 5.5 corresponds to the region where all the three frequencies oscillate in Figure 5.7. This region is identified from 0.4 to 1.4, which corresponds to positive values of first Lyapunov exponent.

Quasi-periodic oscillation is obtained in the other regions of Figure 5.4 which correspond to the region where two of the frequencies oscillate in Figure 5.7. Therefore, we conclude that the condition necessary to generate chaos is three oscillations of the independent voltages across the three junctions. We conjecture that the mechanism of the generation of chaos is a nonlinear frequency mixing among the three oscillating voltages. We observe quasi-periodic oscillations only when two of the voltages oscillate in the other parameter regions. These results are consistent with the origin of chaos caused by the quasi-periodicity-breakdown route.

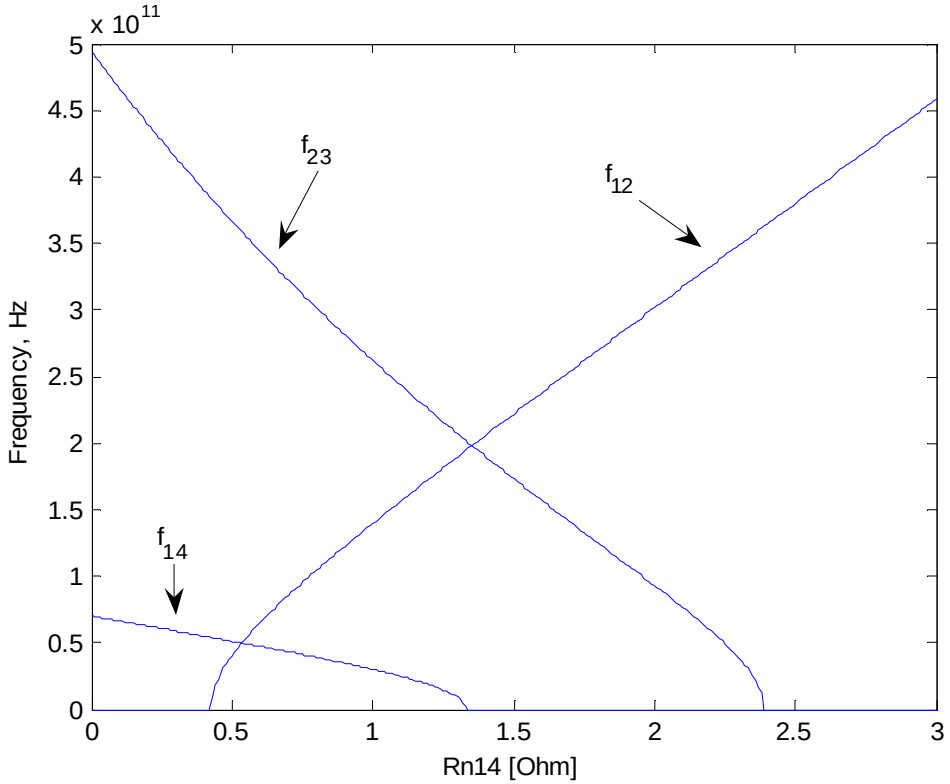


Figure 5.7 Tetrode microbridge electrical voltages frequencies as function of R_{n14}

5.5 Tetrode Microbridge System under Microwave Radiation

In this section we study the Tetrode Microbridge System under microwave radiation, instead of just dc currents, as discussed in a tetrode system. φ_{ij} is the phase difference of wave functions, $R_{n,ij}$ is the normal resistance of the Josephson junction, $I_{c,ij}$ is the critical current in the five Josephson junctions, and $I_1(t)$, $I_2(t)$, $I_3(t)$, are the drive currents including a dc current source and a microwave source with amplitude g and frequency ω_d. $I_1(t) = I_{1dc} + I_{1ac}\ cos(\omega_d t)$, $I_2(t) = I_{2dc} + I_{2ac}\ cos(\omega_d t)$, $I_3(t) = I_{3dc} + I_{3ac}\ cos(\omega_d t)$ and we define once again $i_1 = I_{1dc}(t)/I_{c12}$; $i_2 = I_{2dc}(t)/I_{c12}$; $i_3 = I_{3dc}(t)/I_{c1}$. The notation of the normalized explicit functions of φ_{ij},

is derived from following equations of motion:

$$\frac{\mathrm{d}^2\varphi_{14}}{\mathrm{d}t^2}+\left(\frac{1}{R_{n14}C_{14}}\right)\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}t}+\left(\frac{2\pi I_{c14}}{\Phi_0 C_{14}}\right)\sin\varphi_{14}+\left(\frac{C_{12}}{C_{14}}\right)\frac{\mathrm{d}^2\varphi_{12}}{\mathrm{d}t^2}+\left(\frac{1}{R_{n12}C_{14}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}+$$
$$\left(\frac{2\pi I_{c12}}{\Phi_0 C_{14}}\right)\sin\varphi_{12}=\left(\frac{2\pi}{\Phi_0 C_{14}}\right)I_1(t),$$

$$\frac{\mathrm{d}^2\varphi_{23}}{\mathrm{d}t^2}+\left(\frac{1}{R_{n14}C_{23}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}+\left(\frac{2\pi I_{c23}}{\Phi_0 C_{14}}\right)\sin\varphi_{23}+\left(\frac{C_{24}}{C_{23}}\right)\frac{\mathrm{d}^2\varphi_{24}}{\mathrm{d}t^2}+\left(\frac{1}{R_{n24}C_{23}}\right)\frac{\mathrm{d}\varphi_{24}}{\mathrm{d}t}+$$
$$\left(\frac{2\pi I_{c24}}{\Phi_0 C_{23}}\right)\sin\varphi_{24}-\left(\frac{C_{12}}{C_{23}}\right)\frac{\mathrm{d}^2\varphi_{12}}{\mathrm{d}t^2}-\left(\frac{1}{R_{n12}C_{23}}\right)\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}t}-\left(\frac{2\pi I_{c12}}{\Phi_0 C_{23}}\right)\sin\varphi_{12}=\left(\frac{2\pi}{\Phi_0 C_{23}}\right)I_2(t),$$

$$\frac{\mathrm{d}^2\varphi_{34}}{\mathrm{d}t^2}+\left(\frac{1}{R_{n34}C_{34}}\right)\frac{\mathrm{d}\varphi_{34}}{\mathrm{d}t}+\left(\frac{2\pi I_{c34}}{\Phi_0 C_{34}}\right)\sin\varphi_{34}-\left(\frac{C_{23}}{C_{34}}\right)\frac{\mathrm{d}^2\varphi_{23}}{\mathrm{d}t^2}-\left(\frac{1}{R_{n23}C_{34}}\right)\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}t}$$
$$-\left(\frac{2\pi I_{c23}}{\Phi_0 C_{34}}\right)\sin\varphi_{23}=\left(\frac{2\pi}{\Phi_0 C_{34}}\right)I_3(t).$$

(5.13)

The three independent differential equations for the phase difference of the wave functions φ_{ij}, can be written as follows. The other variances can be described as dependent variances because we assume that there is no magnetic flux in the loop of the Josephson junctions whose length is much smaller than the coherent length. The dynamic system equations are:

$$\eta\frac{\mathrm{d}\varphi_{12}}{\mathrm{d}\tau}=\alpha_{12}+\beta_{12}\sin\varphi_{12}+\gamma_{12}\sin\varphi_{23}+\delta_{12}\sin\varphi_{14}+\sigma_{12}\sin(\varphi_{14}-\varphi_{12})$$
$$+\rho_{12}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23})+g\cos(\omega_d t),$$
$$\eta\frac{\mathrm{d}\varphi_{23}}{\mathrm{d}\tau}=\alpha_{23}+\beta_{23}\sin\varphi_{12}+\gamma_{23}\sin\varphi_{23}+\delta_{23}\sin\varphi_{14}+\sigma_{23}\sin(\varphi_{14}-\varphi_{12})$$
$$+\rho_{23}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23})+g\cos(\omega_d t),$$
$$\eta\frac{\mathrm{d}\varphi_{14}}{\mathrm{d}\tau}=\alpha_{14}+\beta_{14}\sin\varphi_{12}+\gamma_{14}\sin\varphi_{23}+\delta_{14}\sin\varphi_{14}+\sigma_{14}\sin(\varphi_{14}-\varphi_{12})$$
$$+\rho_{14}\sin(\varphi_{14}-\varphi_{12}-\varphi_{23})+g\cos(\omega_d t).$$

(5.14)

The coefficients are defined as before. g is the normalized microwave source amplitude. The introduction of the time-varying term has by its nature changed the tetrode microbridge

autonomous system to the microwave-tetrode non-autonomous system, which is equivalent to a four-dimensional autonomous system. The dynamics of this system can be characterized with its LE, computed numerically by the GSR-based algorithm. This new system will have four Lyapunov exponents, possessing one positive Lyapunov exponent along with one zero and two negative Lyapunov exponents. The plot of the maximum Lyapunov exponents, versus g and ω_d, is obtained separately as shown in Figure 5.8, while the corresponding time-series and phase space attractors are given in Figure 5.9. The control parameters are set to $0 \leq g \leq 1$ with increment 0.01, while the driving microwave radiation frequency is 1, and microbridge junction resistance R_{n14} is 1.4. For quasi periodic orbits $\lambda \approx 0$, while for chaotic trajectories $\lambda > 0$. The first quasioeriodic window occurs for g from 0.12 to 0.18 followed by a chaotic states region from 0.2 to 0.4. As g is increased further from 0.4, the temporal waveforms become quasi-periodic again, and quasi-stable outputs are obtained. The waveforms become more chaotic as g is decreased from 0.12. The temporal waveform fluctuates chaotically in the range 0.12 to 0 where the system turns to the tetrode microbridge autonomous system.

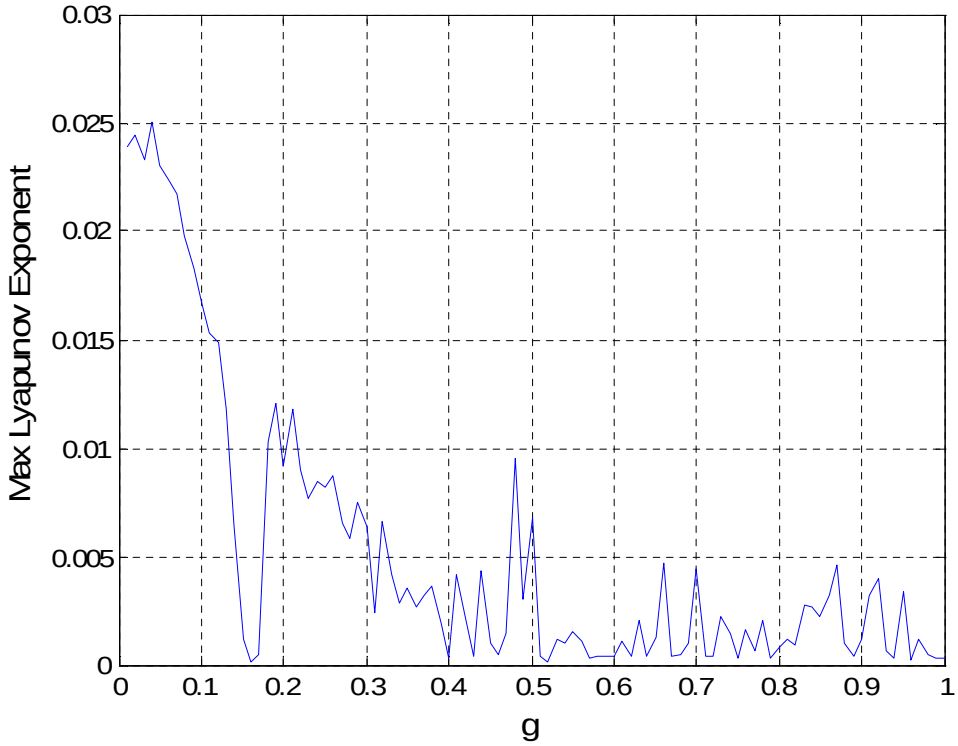


Figure 5.8 Tetrode microbridge under microwave excitation LE. $0 \leq g \leq 1$, $\Delta g = 0.01$, $\omega_d = 1$, $R_{n14} = 1.4$.

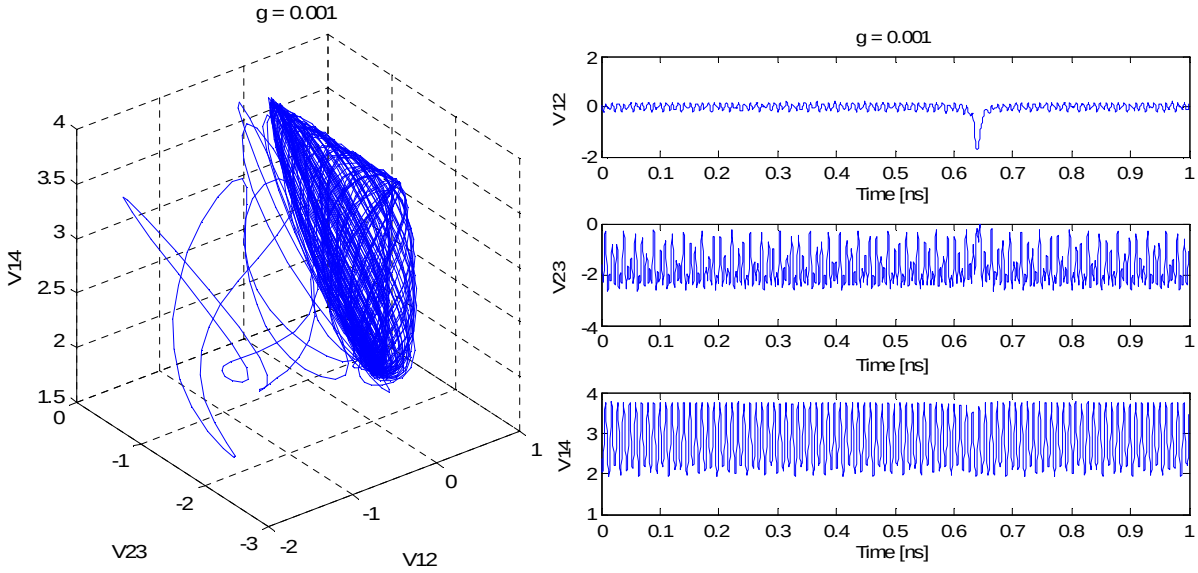


Figure 5.9 Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \leq g \leq 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \leq g < 0.12$, $0.2 \leq g \leq 0.4$. Otherwise the states are quasi-periodic oscillations.

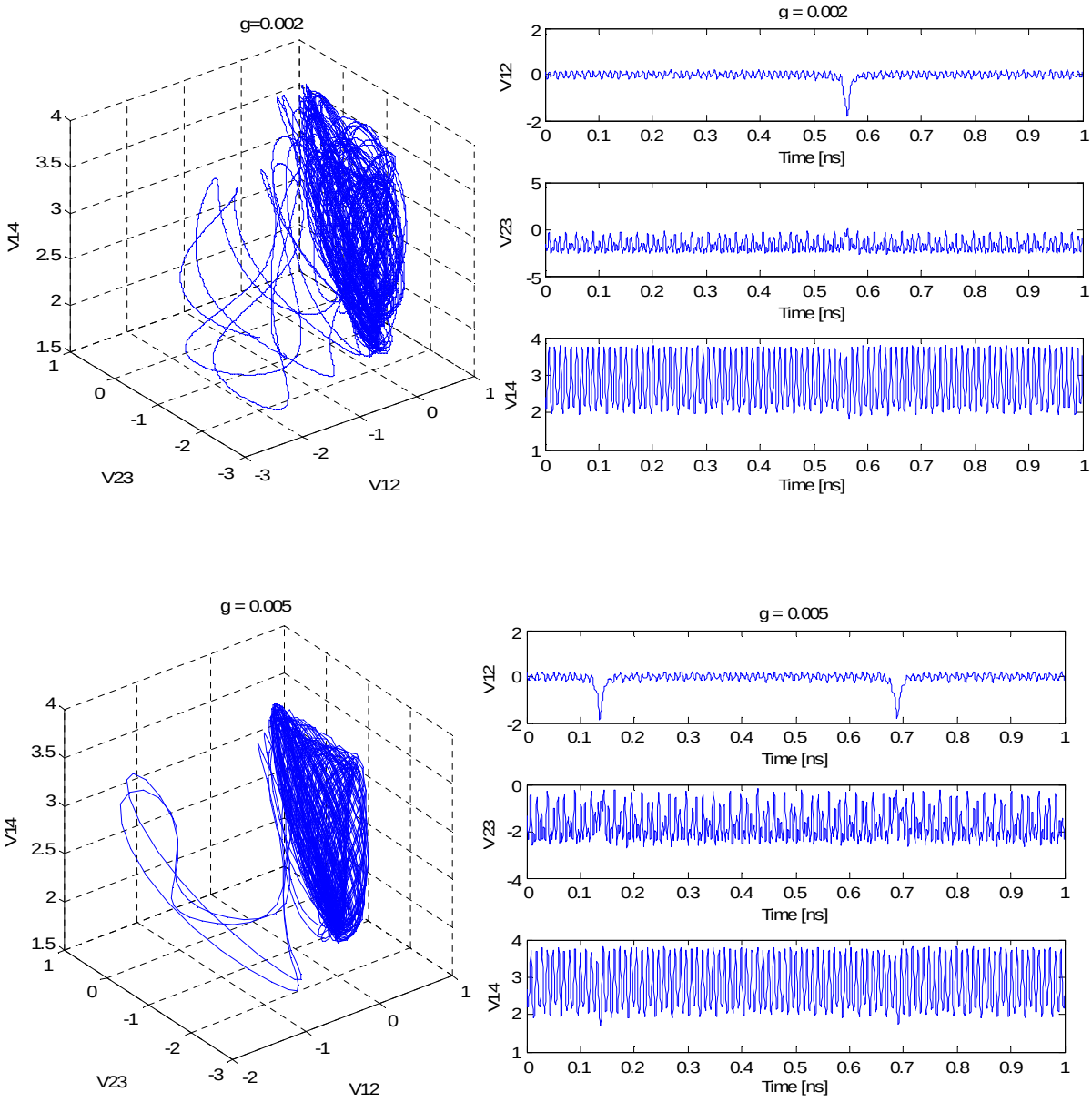


Figure 5.9, cont. Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \leq g \leq 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \leq g < 0.12$, $0.2 \leq g \leq 0.4$. Otherwise the states are quasi-periodic oscillations.

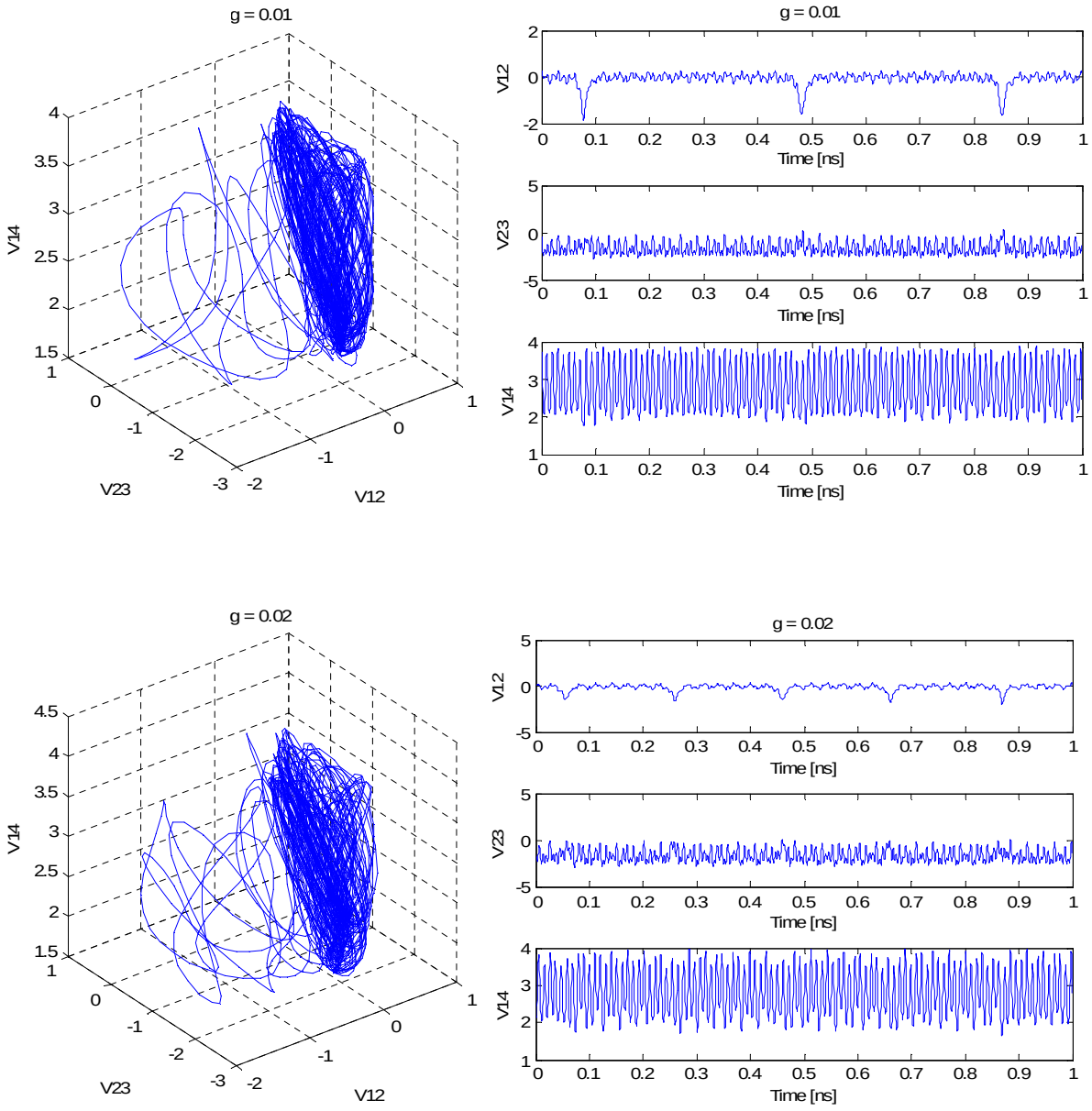


Figure 5.9, cont. Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \leq g \leq 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \leq g < 0.12$, $0.2 \leq g \leq 0.4$. Otherwise the states are quasi-periodic oscillations.

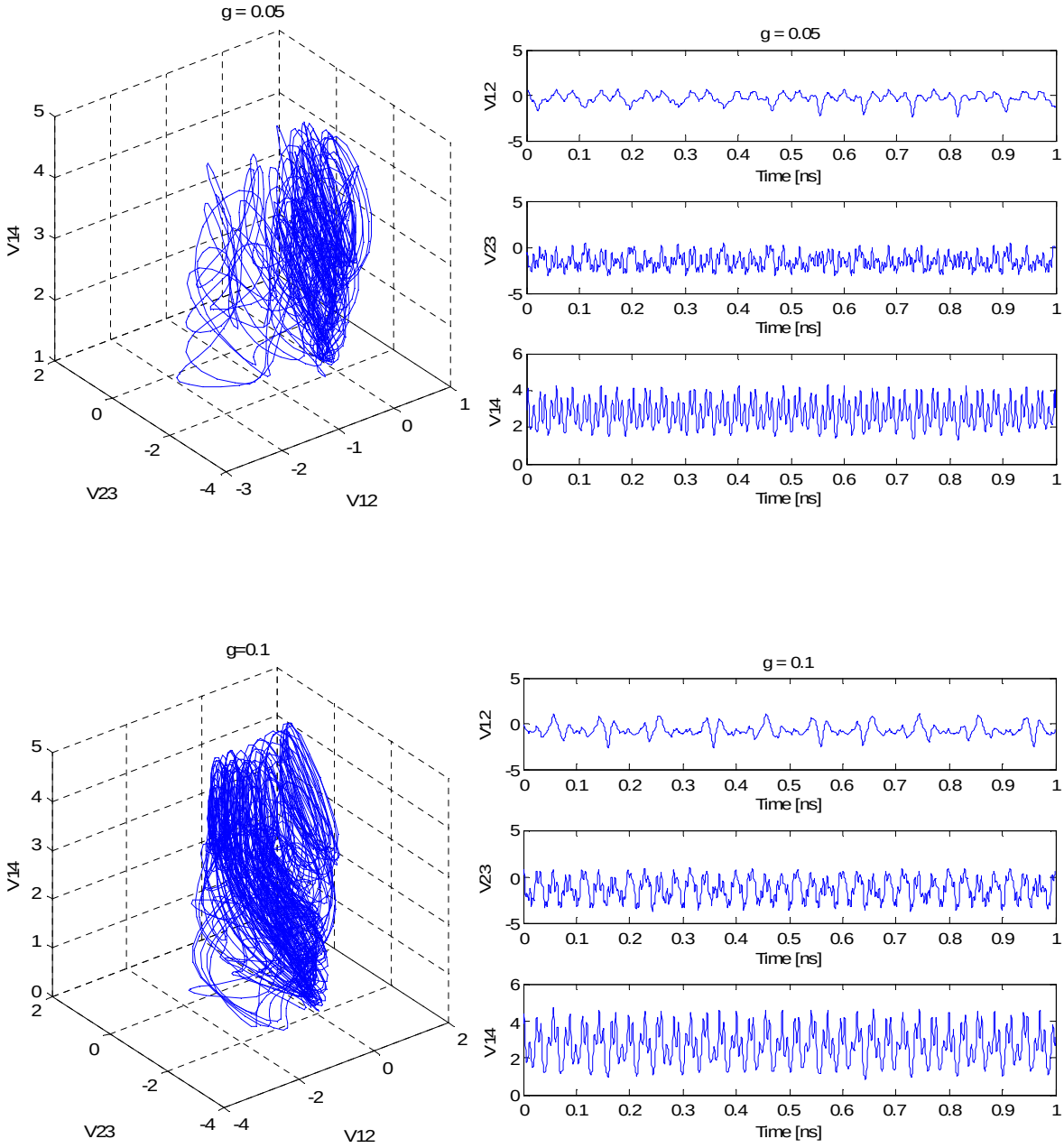


Figure 5.9, cont. Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \leq g \leq 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \leq g < 0.12$, $0.2 \leq g \leq 0.4$. Otherwise the states are quasi-periodic oscillations.

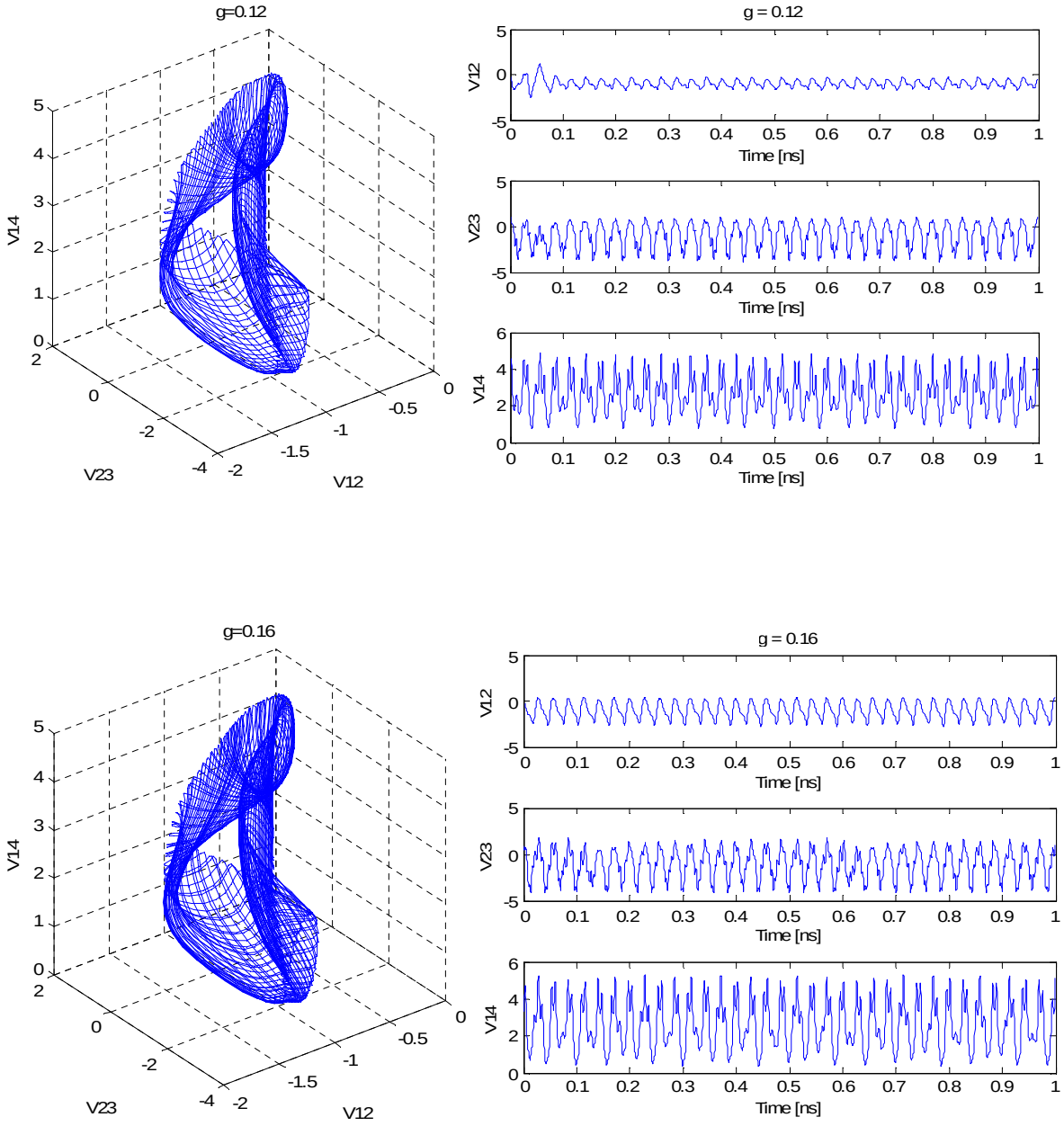


Figure 5.9, cont. Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \le g \le 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \le g < 0.12$, $0.2 \le g \le 0.4$. Otherwise the states are quasi-periodic oscillations.

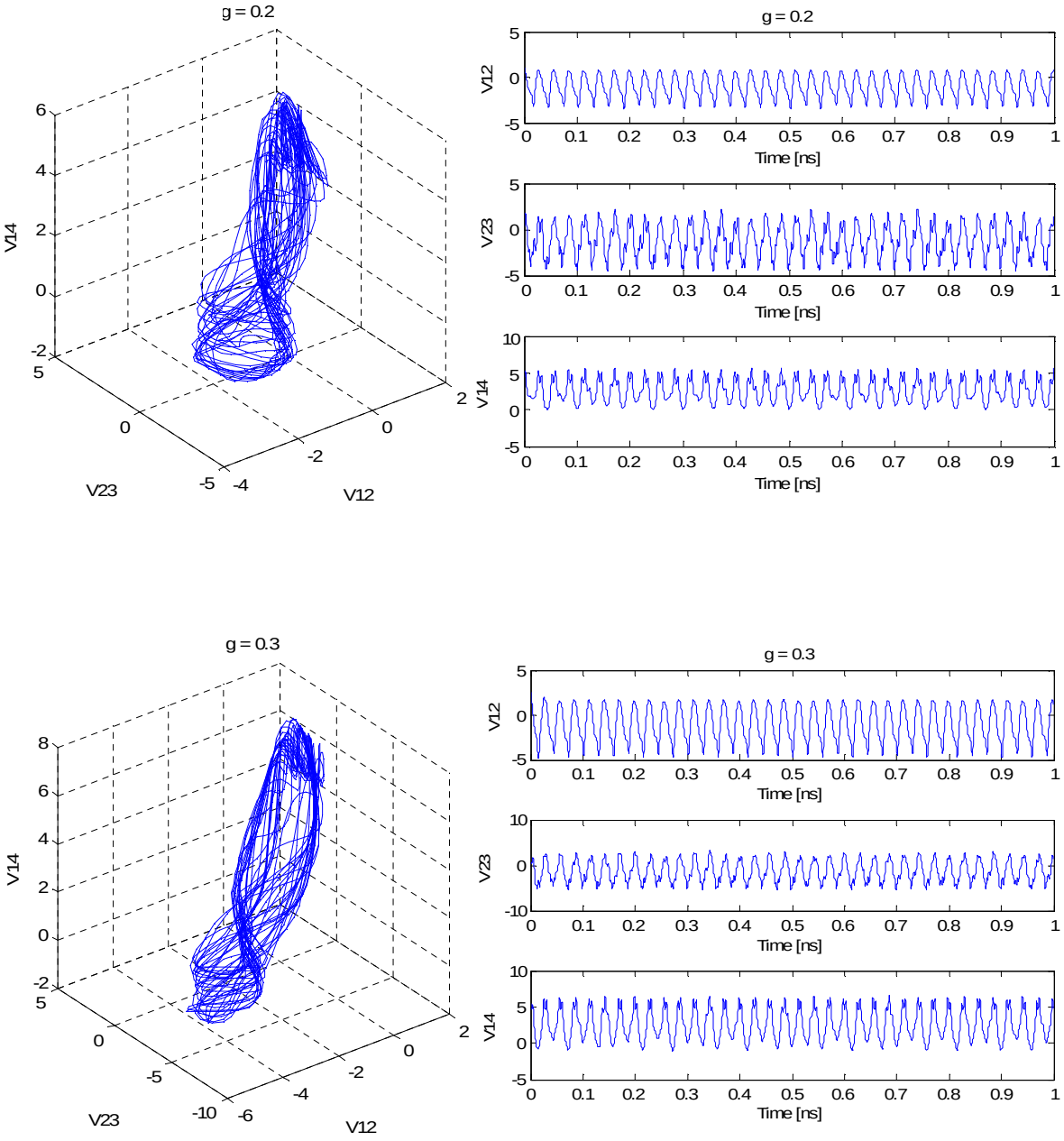


Figure 5.9, cont. Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \leq g \leq 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \leq g < 0.12$, $0.2 \leq g \leq 0.4$. Otherwise the states are quasi-periodic oscillations.

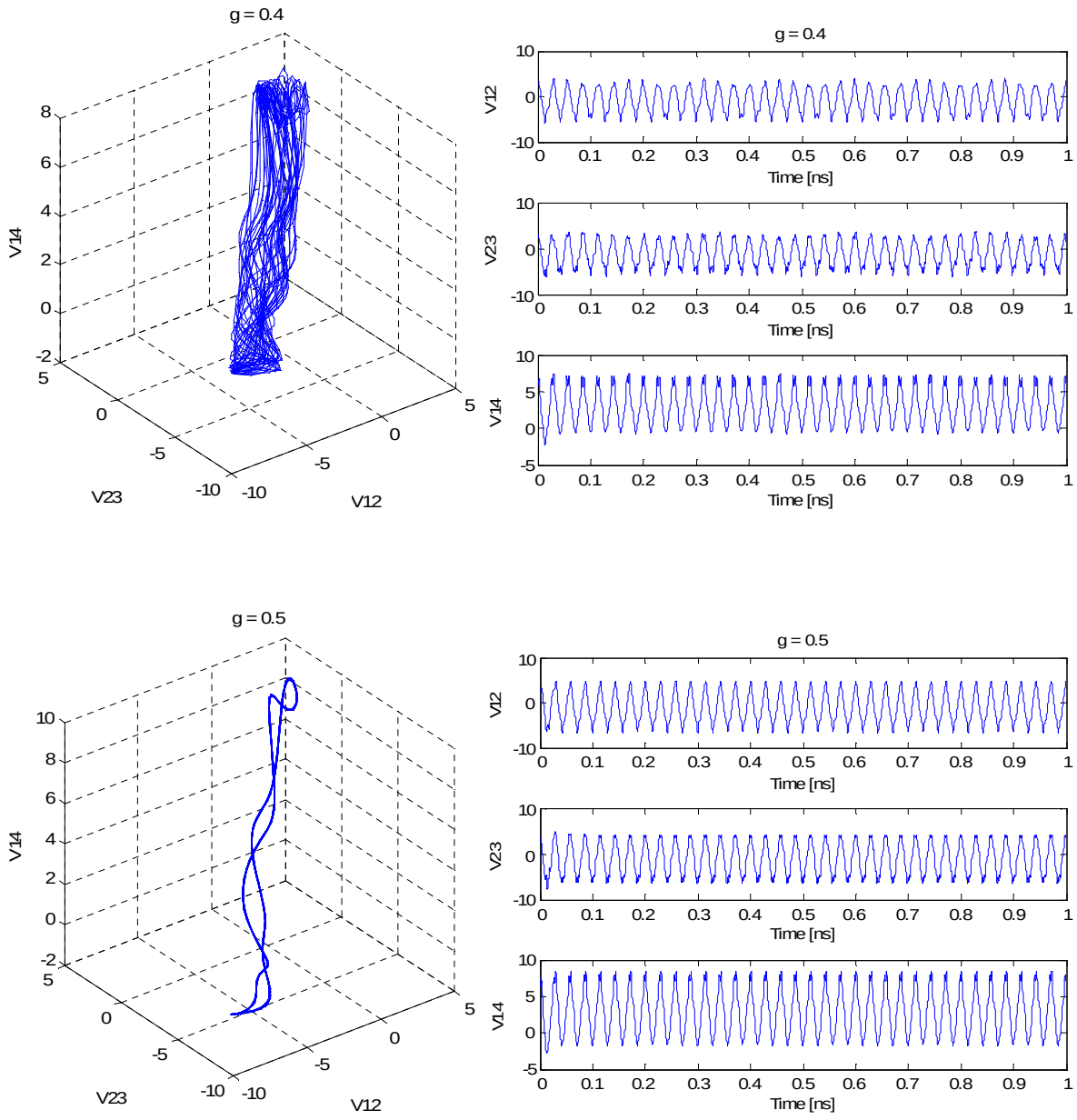
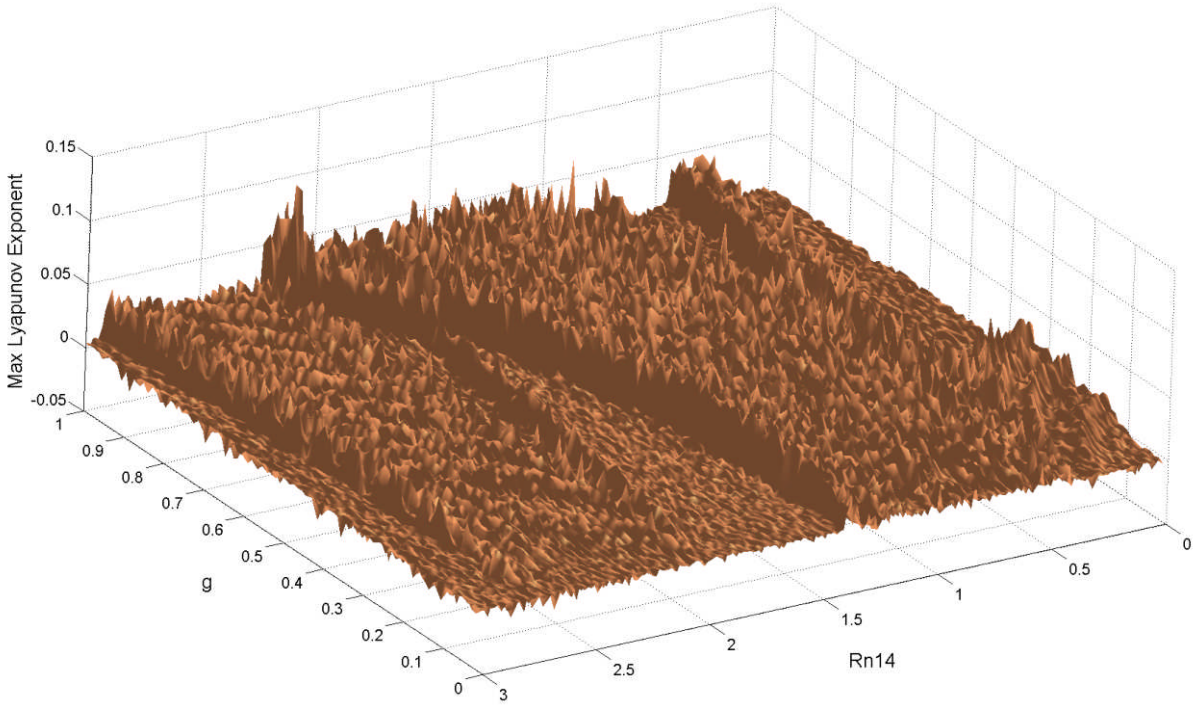


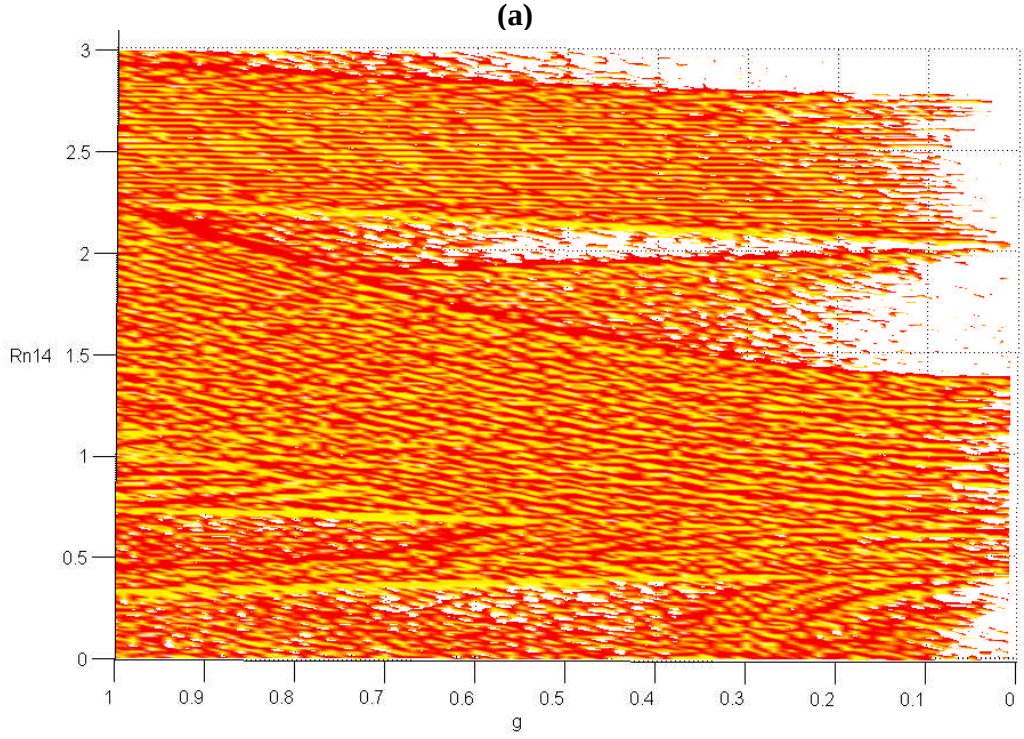
Figure 5.9, cont. Tetrode microbridge under microwave excitation. Electrical voltages portrait and temporal waveforms at various microwave amplitude $0 \le g \le 1$. Microwave frequency $\omega_d = 1$ and $R_{n14} = 1.4\ \Omega$. Chaotic oscillations correspond to $0 \le g < 0.12$, $0.2 \le g \le 0.4$. Otherwise the states are quasi-periodic oscillations.

To determine the dependency of the system states on the parameters g, R_{n14}, and ω_d we draw up the parametric phase transition diagram for the tetrode microbridge under microwave excitation. We use two of the control parameters in a grid with the initial conditions $\varphi_{12} = 1.2$,

φ_{23} = 1.4, φ_{14} = 1.1. The phase diagram is drawn in the range from 0 to 1 for the parameter g with Δg = 0.01, and from 0 to 3 Ω for R_{n14}, with ΔR_{n14} = 0.01. This means the region of phase is subdivided into 100 by 300 cells. Driving frequency is ω_d = 1. Each Lyapunov exponent computation is performed over 10^5 iterations. 10^4 transient solutions are discarded with integration time step ε = 0.01. We construct the phase diagram of the parameter space by showing the positive values of the first Lyapunov exponent. This is depicted in Figure 5.10 (a). In Figure 5.10 (b) the positive maxima of the Lyapunov exponent λ^+ are identified. There are regions of quasiperiodic states in wide chaotic states. When g = 0.97, R_{n14} = 2.15 Ω, the positive Lyapunov exponent has its absolute maximum of 0.16.



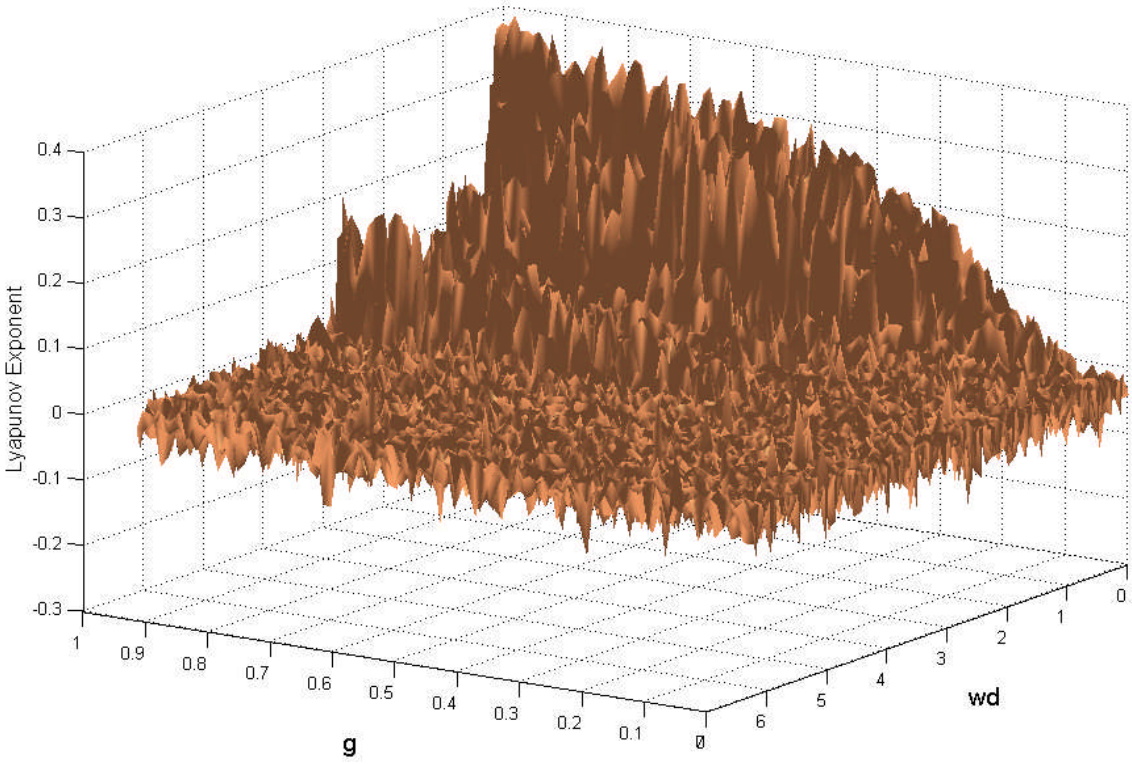
(a)



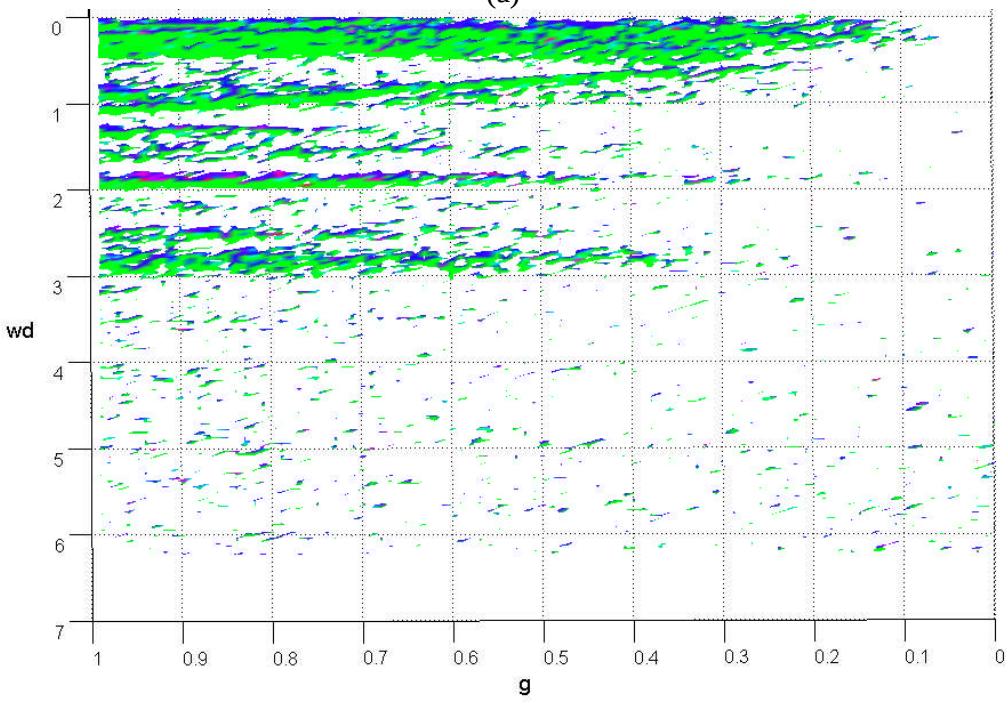
(b)

Figure 5.10 Tetrode microbridge under microwave excitation. (a) first LE λ_1 (b) first positive maximum Lyapunov exponent λ_1^+. Control parameters setting: drive amplitude $0 \leq g \leq 1$ [Δg = 0.01], and $0 \leq R_{n14} \leq 3$ [ΔR_{n14} =0.01], microwave frequency is ω_d = 1. First LE maximum value is 0.16 and occurs when g = 0.97, R_{n14} = 2.15.

In another trial which is illustrated in Figures 5.11 (a) and 5.11 (b), the dynamical transitions from quasiperiodic to chaotic states are also traced by varying ω_d in the parameter range $0 \leq \omega_d \leq 7$, with increment $\pi/100$ and drive amplitude g in $0 \leq g \leq 1$ with increment 0.01. R_{n14} is 1.4. Also by exploring the positiveness of the first Lyapunov exponent, regions of chaotic responses are observed. The positive first Lyapunov exponent of $\lambda_1^+ = 0.398$ occurs at $g = 0.96$ and $\omega_d = 0.25$. The phase portrait for this particular state is shown in Figure 5.12. Samples of phase portraits and temporal voltages waveforms for various control parameters setting are provided. When the control parameters setting are $\omega_d = \pi/10$, $R_{n14} = 0$, Figure 5.13 shows two quasiperiodic states at $g = 0.1$, 0.2, and a chaotic state at 0.3. And Figure 5.14 presents the quasiperiodic states when drive amplitude and frequency are $g = 0.1$, $\omega_d = \pi/10$, for R_{n14} values equal to 0.4, 0.5, 0.7.



(a)



(b)

Figure 5.11 Tetrode microbridge under microwave excitation. (a) first LE λ_1 (b) positive maximum LE λ_1^+. Control parameters: microwave amplitude $0 \leq g \leq 1$ [$\Delta g = 0.01$], microwave frequency $0 \leq \omega_d \leq 7$ [$\Delta\omega_d = \pi/100$], R_{n14}= 1.4 Ω. Maximum positive first LE, $\lambda_1^+ = 0.398$ occurs at $g = 0.96$ and $\omega_d = 0.25$. The phase portrait for this attractor is shown in Figure 5.12.

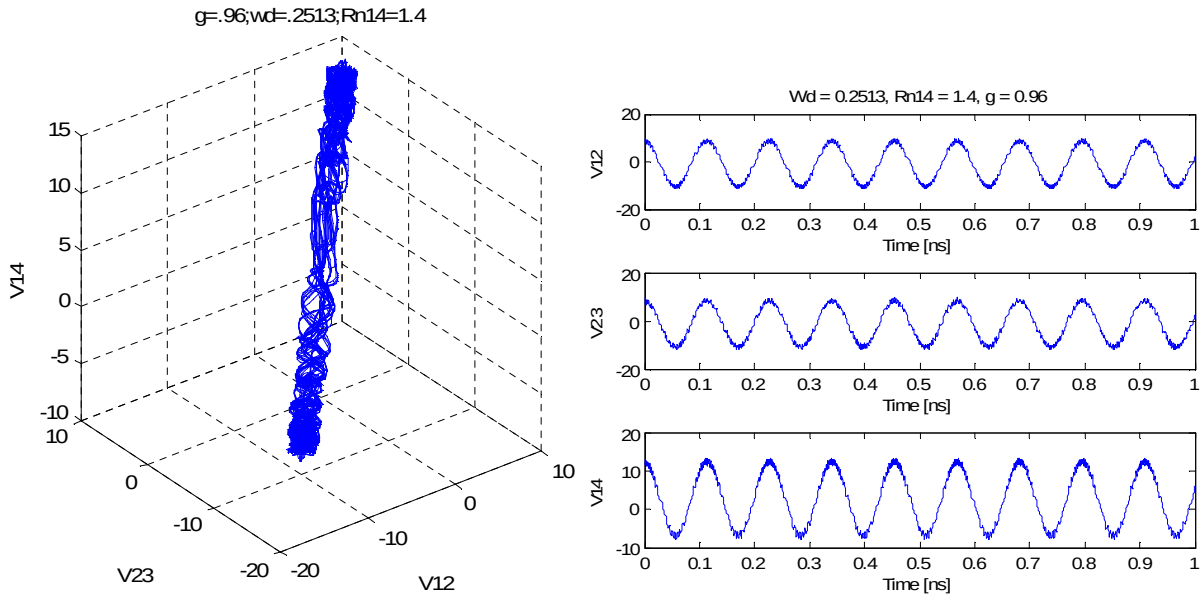


Figure 5.12 Tetrode microbridge under microwave excitation phase portrait. The maximum positive first Lyapunov exponent ($\lambda_1^+ = 0.3983$) at $g = 0.96$ and $\omega_d = 0.2513$.

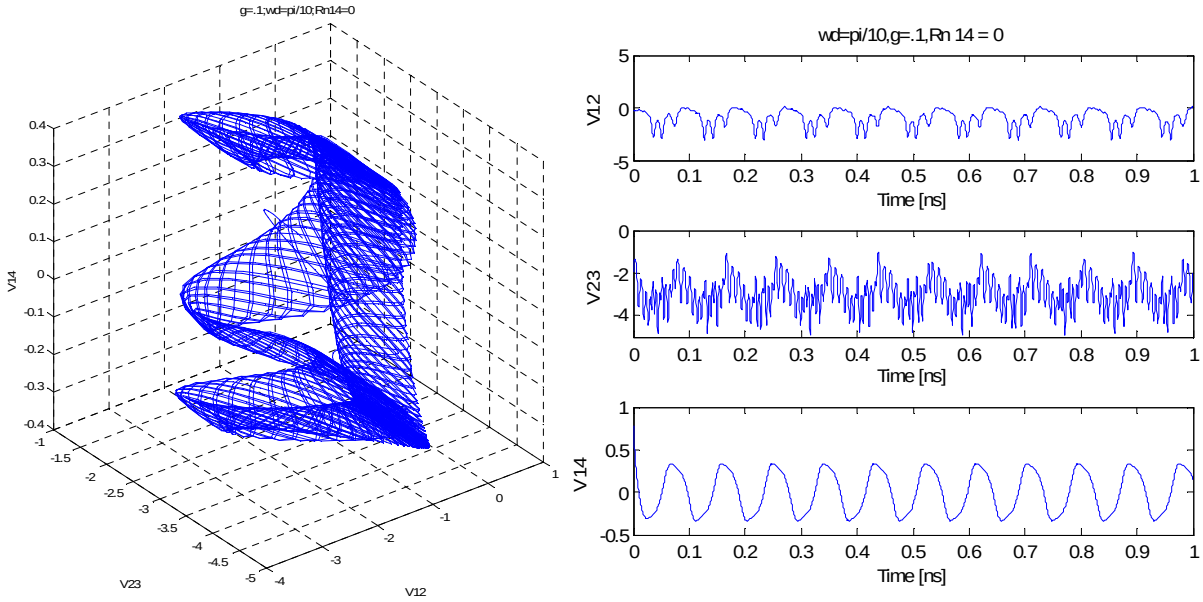


Figure 5.13 Tetrode microbridge under microwave excitation. Voltage phase portraits and temporal waveforms. Control parameters: $\omega_d = \pi/10$, $R_{n14} = 0$. Showing quasiperiodic states at $g = 0.1$, 0.2 and a chaotic state at 0.3.

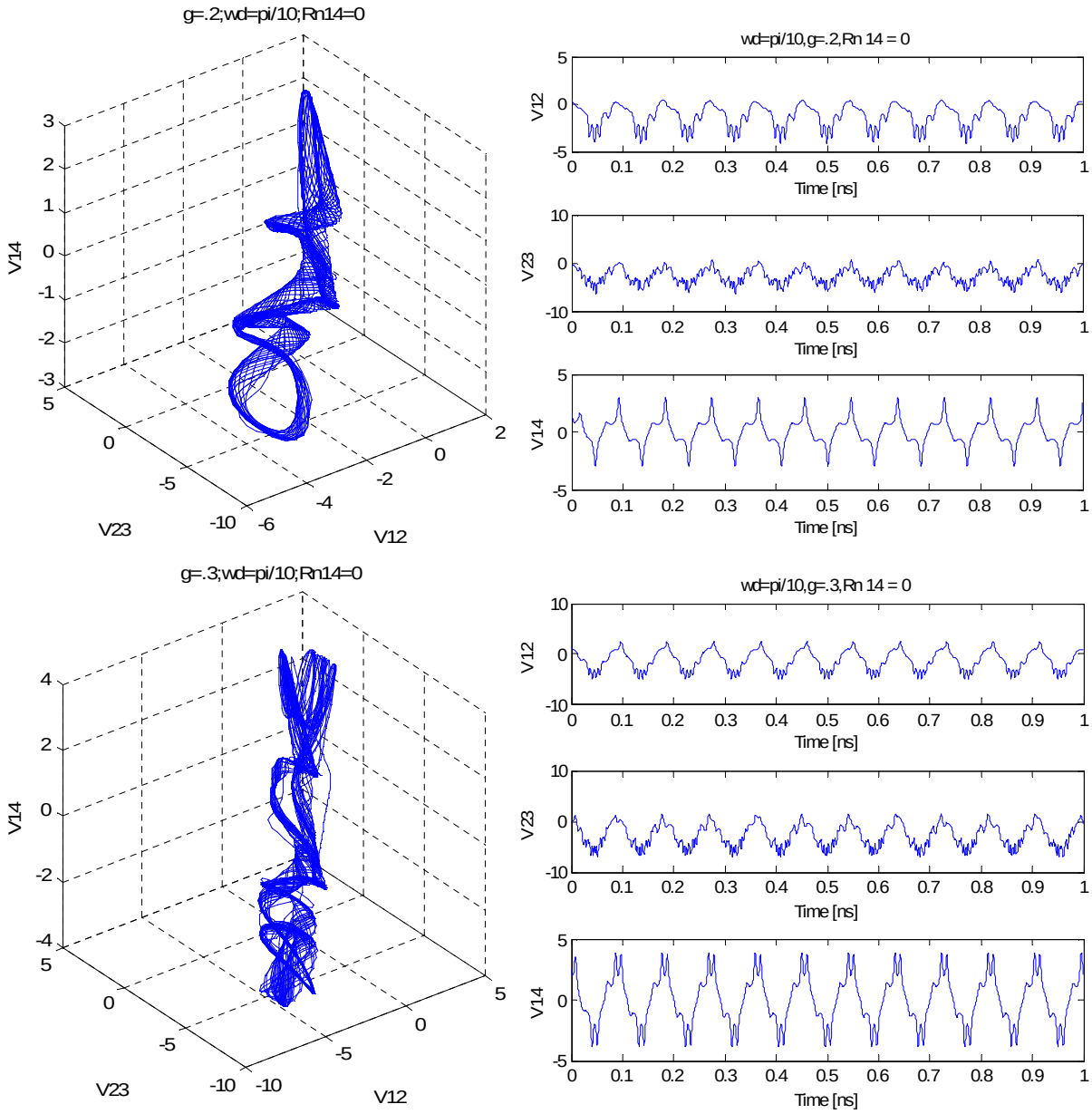


Figure 5.13, cont. Tetrode microbridge under microwave excitation. Voltages phase portraits and temporal waveforms. Control parameters: $\omega_d = \pi/10$, $R_{n14} = 0$. Showing two quasiperiodic states at $g = 0.1$, 0.2 and a chaotic state at 0.3.

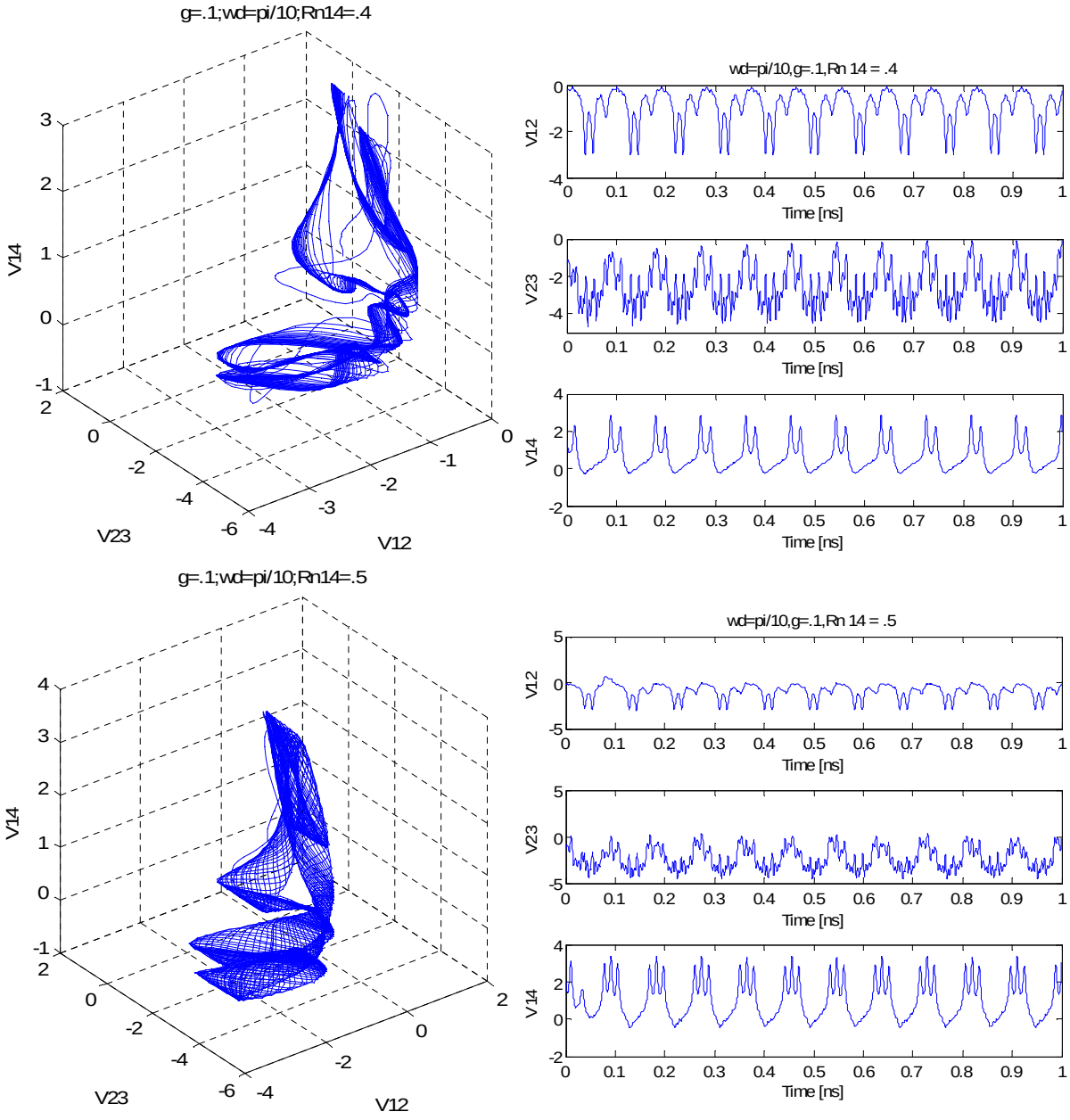


Figure 5.14 Tetrode microbridge under microwave excitation phase portraits and temporal voltages waveforms. Control parameters setting for microwave amplitude and frequency are g = 0.1, $\omega_d = \pi/10$. R_{n14} values are 0.4, 0.5, 0.7 Ω.

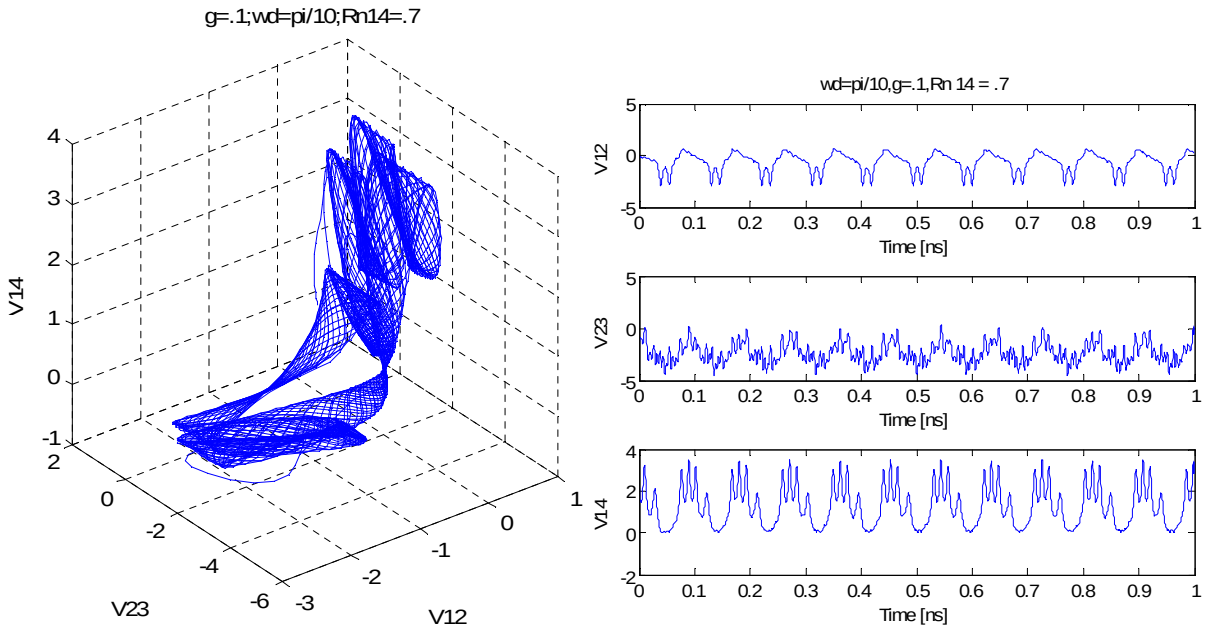


Figure 5.14, cont. Tetrode microbridge under microwave excitation phase portraits and temporal voltages waveforms. Control parameters setting for microwave amplitude and frequency are g = 0.1, $\omega_d = \pi/10$. R_{n14} values are 0.4, 0.5, 0.7 Ω.

CHAPTER 6

FEEDBACK-CONTROLLED HYPERCHAOS

6.1 Introduction to Hyperchaos

The occurrence of two or more positive Lyapunov exponents is called hyperchaos. A hyperchaotic system has the characteristics of high complexity, and hence high security and efficiency in data communication and storage. In an N-dimensional phase space, there are N Lyapunov exponents. Each exponent measures the relative stretching and contraction of the principle axes of the small element of initial conditions. In many systems, this stretching occurs along one direction and results in one positive Lyapunov exponent. Hyperchaos occurs because in some systems, two or more directions in the phase space suffer stretching under the dynamic process and this result in two or more positive Lyapunov exponents. Therefore, to generate hyperchaotic states from the dissipative autonomous polynomial systems, the state equation must satisfy the following basic conditions. The dimension of the state equations should be at least four and the order of the state equations is at least two. In addition, the system has at least two positive Lyapunov exponents satisfying the conditions that the sum of all Lyapunov exponents is less than zero. Hence, hyperchaos does not occur in Hamiltonian systems. Typical examples are the 4-dimensional hyperchaotic Rössler system, the hyperchaotic Lorenz–Haken system, the hyperchaotic Chua's circuit, the hyperchaotic Chen system and the 5-dimensional coupled Van der Pol system [95, 96, 97, 98].

6.2 Feedback-Controlled Hyperchaotic System

This section introduces a new hyperchaotic system with a state feedback controller. This system has four state variables. We investigate the dynamical behaviors of this system for possible hyperchaotic states. Numerical simulation of dynamical states of this system are investigated and compared by means of the Lyapunov exponent spectrum and LZC.

$$\begin{aligned}
\frac{dx_1}{dt} &= A(x_2 - x_1) + x_4 \\
\frac{dx_2}{dt} &= -x_1 x_3 + C x_2 \\
\frac{dx_3}{dt} &= x_1 x_2 - B x_3 \\
\frac{dx_4}{dt} &= x_1 x_3 + D x_4
\end{aligned} \tag{6.3}$$

It is observed that the Lyapunov exponents spectrum of the system (Eqs. 6.3) are (λ_1, λ_2, λ_3, λ_4), for four directions, satisfying $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4$. Figure 6.1 shows the plot of the Lyapunov exponent spectrum for the 4-dim feedback controlled system. The control parameters B and C are set to $B = 3$, $C = 20$, while D varies from 30 to 39 and is incremented by $\Delta D = 0.01$. the parameter A varies from -1.5 to 1.5, with step $\Delta A = 0.01$. The system equations (6.3) are integrated with time-step 0.001, over 2×10^5 iterations, where 2×10^4 transient states starting from the initial state, $(x_1, x_2, x_3, x_4) = (-0.05, 1, -1, 1.2)$, are discarded.

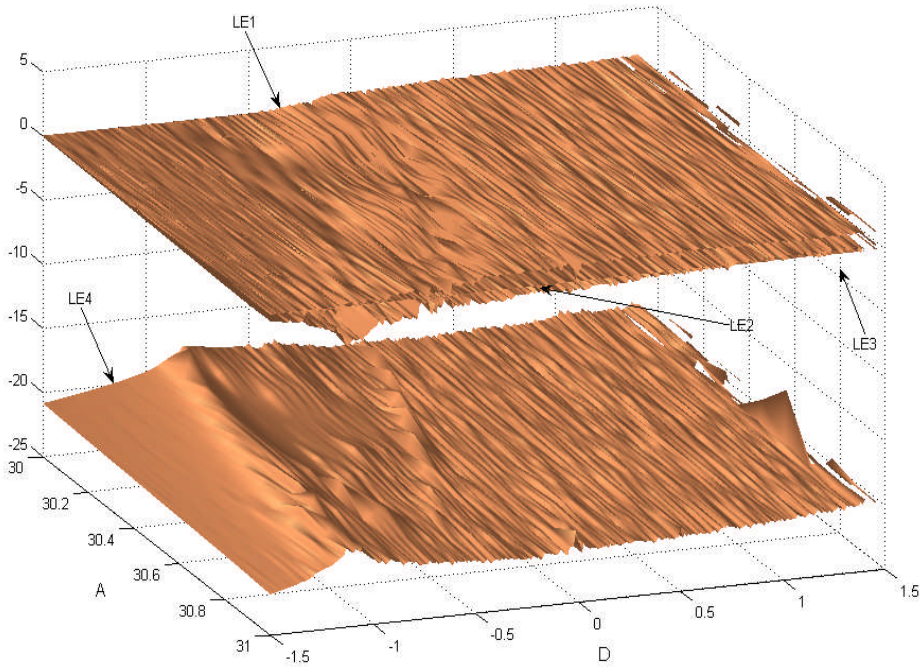


Figure 6.1 Lyapunov exponents spectrum for a feedback-controlled system when $B = 3$, $C = 20$, $30 \leq A \leq 31$ [$\Delta A = 0.01$] and $-1.5 \leq D \leq 1.5$ [$\Delta D = 0.01$].

The dynamical behaviors of this system (Eqs. 6.3) can be classified as follows. For λ_1, λ_2, λ_3, λ_4 the system is hyperchaotic if $0 < \lambda_2 < \lambda_1$, $\lambda_3 = 0$, $\lambda_4 < 0$, and $\lambda_1 + \lambda_2 + \lambda_4 < 0$. The system is chaotic if $\lambda_1 > 0$ and $\lambda_2 = 0$, $\lambda_4 < \lambda_3 < 0$ and $\lambda_1 + \lambda_3 + \lambda_4 < 0$. The system is periodic if $\lambda_1 = 0$ and $\lambda_4 < \lambda_3 < \lambda_2 < 0$. Finally, when all the Lyapunov exponents are negative and satisfy the condition $\lambda_4 < \lambda_3 < \lambda_2 < \lambda_1 < 0$, system has an equilibrium point. The hyperchaotic and chaotic regions are determined from positive values of the first and second Lyapunov exponents. Figure 6.2 shows these exponents. Their respective projections on positive the plane in terms of control space parameters is provided in Figure 6.3 (a) for chaotic plus hyperchaotic regions. Figure 6.3 (b) shows only the hyperchaotic states region.

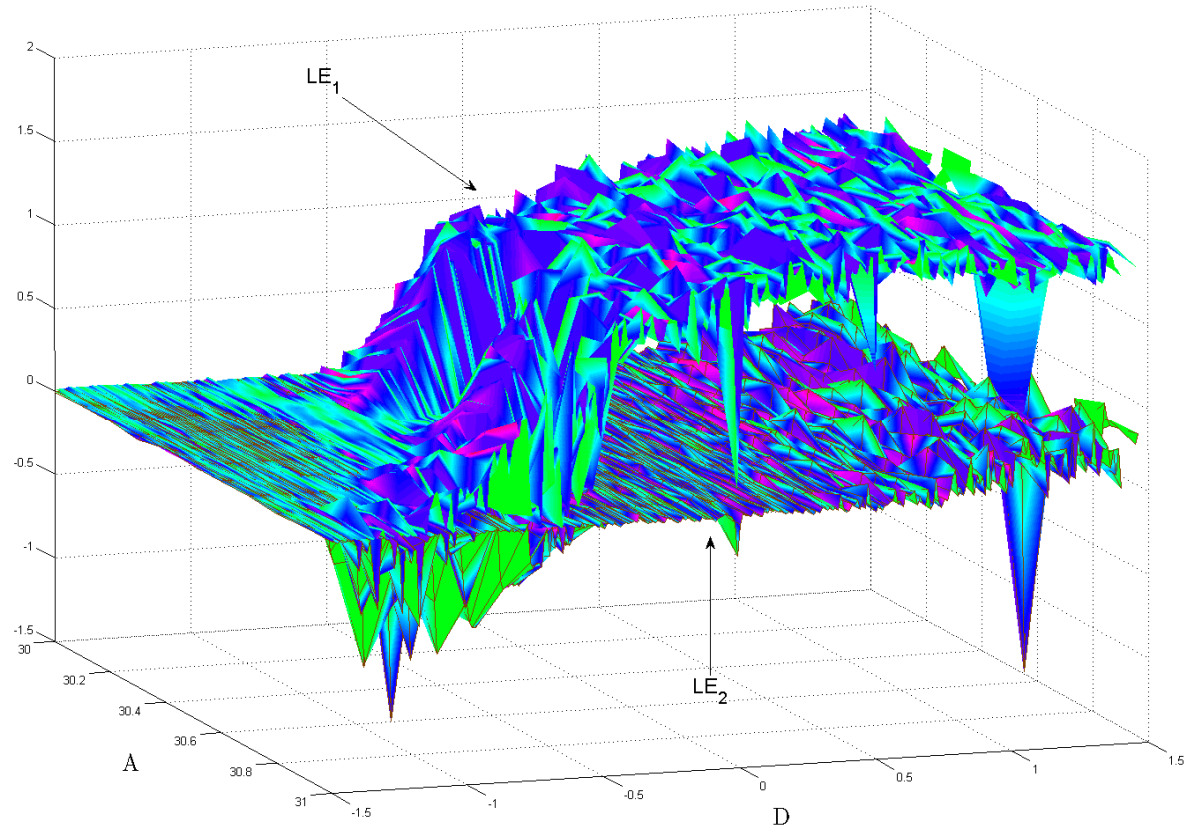
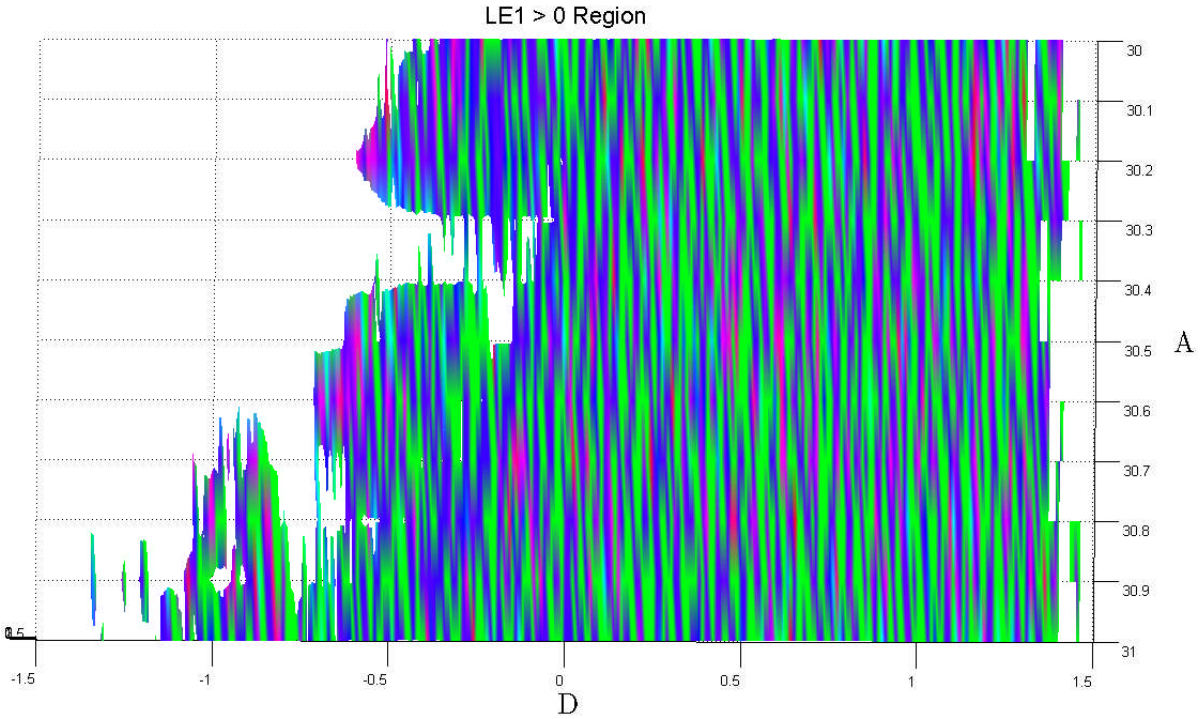
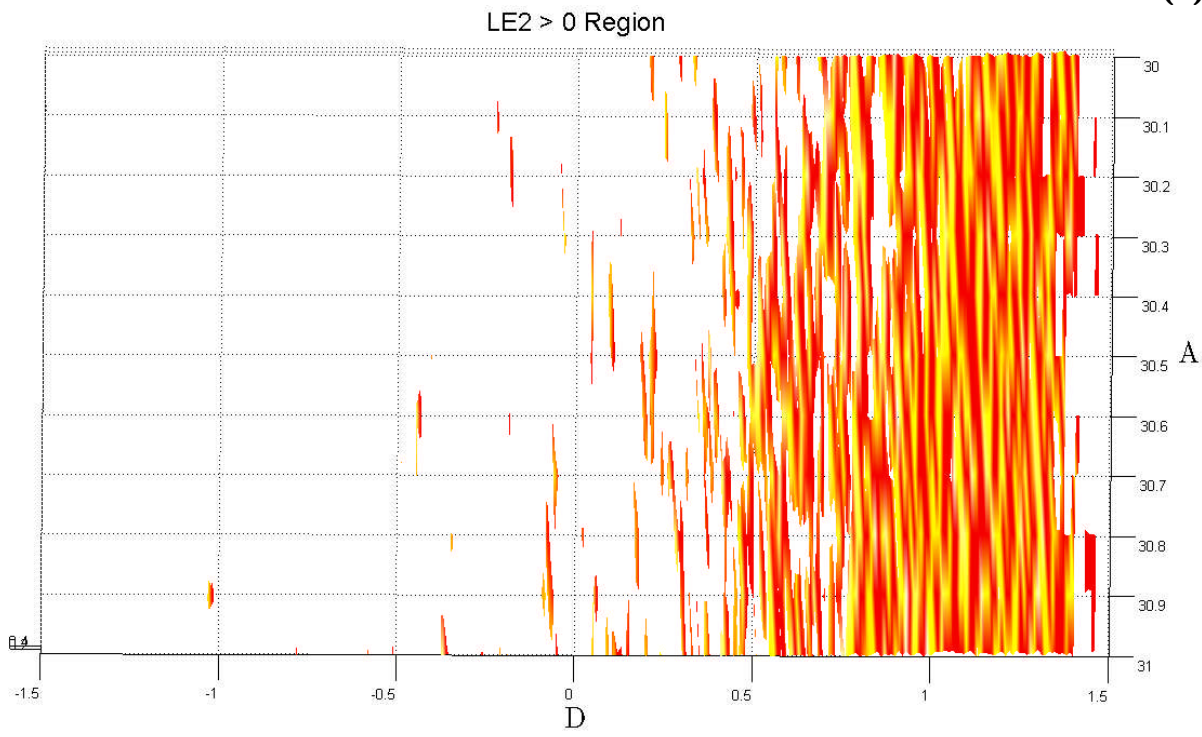


Figure 6.2 First and second Lyapunov exponent plots for 4-dim feedback-controlled system.



(a)



(b)

Figure 6.3 Chaos and hyperchaos regions for 4-Dim feedback-controlled system defined by (Eqs. 6.3) (a) chaos and hyperchaos region. (b) hyperchaos region

Moreover, the Lyapunov fractal dimension of the system (Eqs. 6.3) satisfies $2 < d_f < 3$ for the chaos case and $3 < d_f < 4$ for the hyperchaos case as shown in Figure 6.4. Our numerical analysis shows that the dynamical behaviors of the system switch among chaotic state, periodic orbit, and hyperchaotic state with the increase of control parameters *A* and *D*.

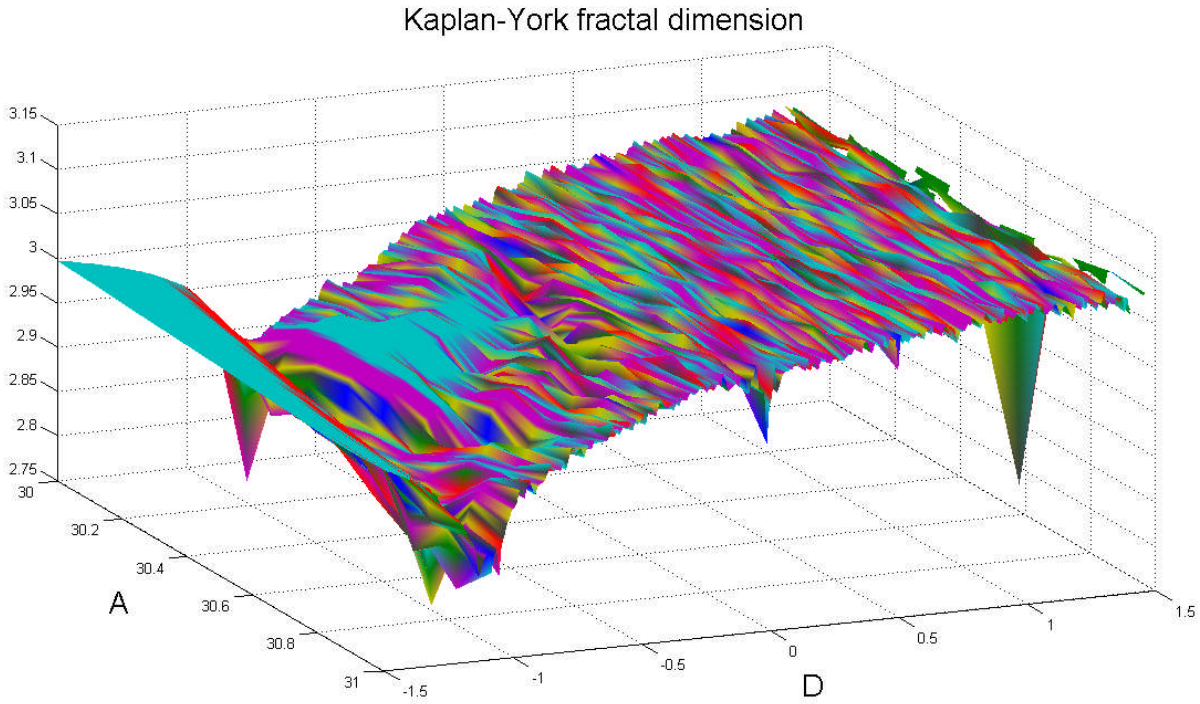


Figure 6.4 Kaplan-York fractal dimension vs. parameters *A* and *D*.

To simplify future analysis the LE spectrum for a specific values of $A = 36$, $B = 3$ and $C = 20$, when *D* varies from –1.5 to 1.2 with step 0.01 is selected and illustrated in Figure 6.5. Periodic behaviors are for $-1.03 \leq D \leq -0.46$. Chaotic attractors occur about $-0.46 < D \leq -0.2$, and a hyperchaotic response for $-0.2 \leq D \leq 1.2$. For $1.2 \leq D \leq 1.5$, there are some hyperchaotic attractors while for some values of *D*, the system diverges, and hence there is no bound solution. This diagram covers completely the real parameter region of *D*.

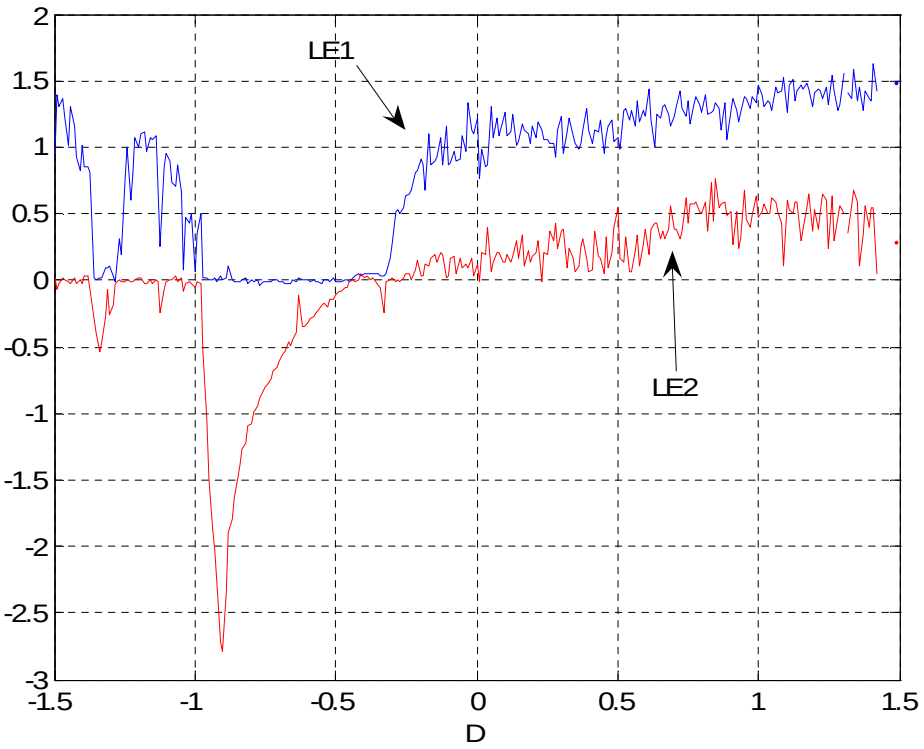


Figure 6.5 Lyapunov exponents λ_1, λ_2 for D: $-1.5 \leq D \leq 1.5$ [$\Delta D = 0.01$], A=36, B=3 and C=20 over 10^5 iterations with 2×10^4 transients, ε=0.001, GSR step 1.

To determine the eigenvalues of the Jacobian matrix $\boldsymbol{J}$ for stability analysis, the system characteristic equation $\det(\boldsymbol{J}-\lambda\boldsymbol{I}) = 0$ is developed and solved. The solutions provide three equilibrium points if the condition $C-AD \neq 0$ is satisfied. These points are

$$Q_0(0,0,0,0)$$

$$Q_1\left(\sqrt{BC}, -\frac{AD\sqrt{BC}}{C-AD}, -\frac{ACD}{C-AD}, \frac{AC\sqrt{BC}}{C-AD}\right),\ Q_2\left(-\sqrt{BC}, \frac{AD\sqrt{BC}}{C-AD}, \frac{ACD}{C-AD}, -\frac{AC\sqrt{BC}}{C-AD}\right) \tag{6.4}$$

The corresponding eigenvalues for point O when A = 36, B = 3, C = 20, D = 1.2, are 20, 1.2, −3, −36. Point Q_0 is a two-dimensional unstable saddle point. This is an important point in the LZ complexity analysis of phase space attractors. The eigenvalues belonging to Q_1 and Q_2 for the

above system parameters are 0.73, −14.16, −2.13 ± j23.11. Therefore, these points are one dimensional unstable saddle points. The latter equilibrium points are symmetric about the x_1, x_2, x_4 axes for any control parameter value.

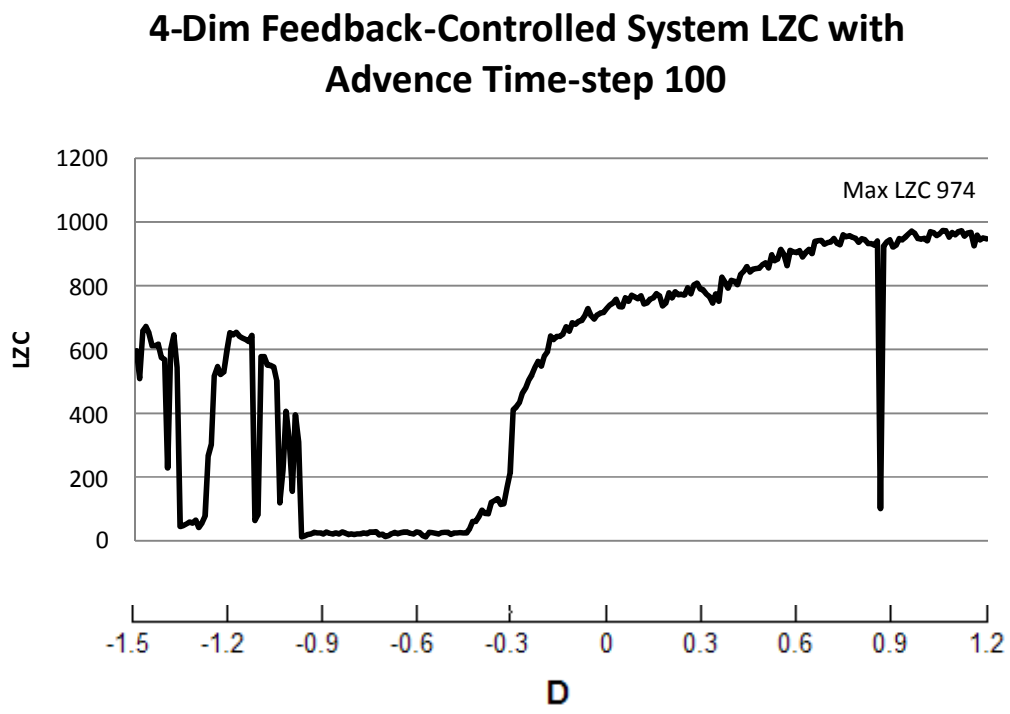
6.3 Computing Lempel-Ziv Complexity by Advance-Time Sampling Method

In this section for slow dynamic systems, the Advance-Time Sampling (ATS) method is introduced and implemented algorithmically to overcome the scarcity of phase space samples in algorithmic complexity measure estimation in feedback controlled hyperchaotic systems. Simulations are provided as evidence that this technique reduces complexity estimation error.

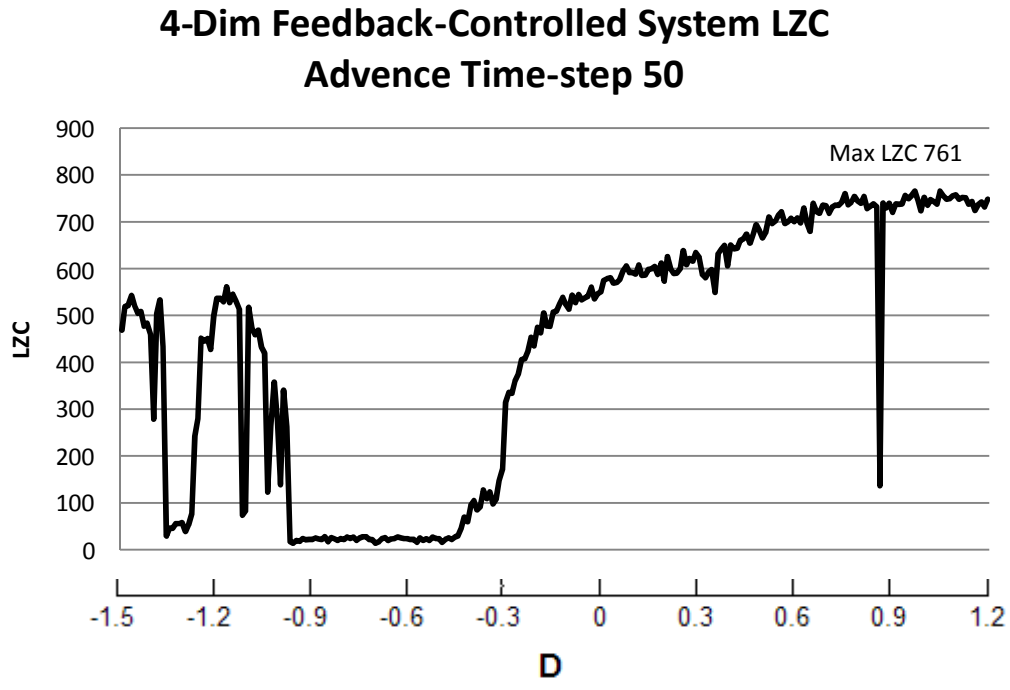
Our simulation results shows that the rate of information generation in some high dimensional systems such as 4-dim feedback-controlled systems (Eqs. 6.3) is too low for a given integration time-step. The slowness of the system is defined based on the scarcity of phase space samples when the system evolves with the integration time-step. This causes longer processing time, as well as larger memory and buffer capacity requirements. Thus, for these systems, immediate solutions are not good candidates for sampling and mapping to binary sequences. To overcome this problem and eliminate the need for longer memory size, ATS is designed. In this method instead of using real-time dynamic states, advance solutions are selected. This means, if trajectories traced and sampled with advance time τ, then system dynamics provides adequate information from phase space attractors for mapping to binary value phase space. The results of two different advance-time gaps are shown in Figure 6.6 (a) and (b). The computation is performed based on the following decision algorithm. But first the projection of phase space on the x_1-x_3 sub-space is partitioned into four regions: ($x_1 \geq 0$; $x_3 \geq 20$), ($x_1 \leq 0$; $x_3 \geq 20$), ($x_1 \leq 0$; $x_3 \leq 20$), ($x_1 \geq 0$; $x_3 \leq 20$). After removal of 5×10^4 transient responses, when the trajectory lands in

zone 1 and 3, bit 1 is inserted into the sequence with length 16300; otherwise bit 0 is inserted. The integration is performed with time step 0.001 for specific values of control parameters set to $A = 36$, $B = 3$, $C = 20$, when D is varied from –1.5 to 1.2 with step 0.01. Periodic behaviors are for $-1.03 \leq D \leq -0.46$ as well as some narrow windows. Chaotic and hyperchaotic attractors occur at approximately $-0.46 < D \leq 1.2$. Example trials of phase portraits and time series choosing hyperchaotic D=0.75, chaotic $D = -1.5$ and periodic $D = -0.75$ states are shown in Figure 6.7.

One interesting unique behavior that we found is that in a 4-dim feedback controlled hyperchaotic system the Lempel-Ziv complexity has a non-decreasing characteristic envelope behavior just before the occurrence of attractor collapse when a diverging dynamic state occurs for $D > 1.2$. In chaotic regions, the LZC measure increases from low values to high values and then drops in periodic windows, but does not have a non-decreasing envelope. We believe this is a characteristic signature of the hyperchaotic systems. We observed this behavior in other hyperchaotic systems when their dynamics investigated by a LZC measure [99]. Another interesting observation is that the hyperchaos region starts when the LZC reaches its equivalent LZC value in the chaotic region. We spot this around $D = 0.2$. In addition, we notice that the graph of LZC is shape preserved but the scale is larger for larger advance time-step ($\tau = 100$) compared to the smaller choice ($\tau = 50$).

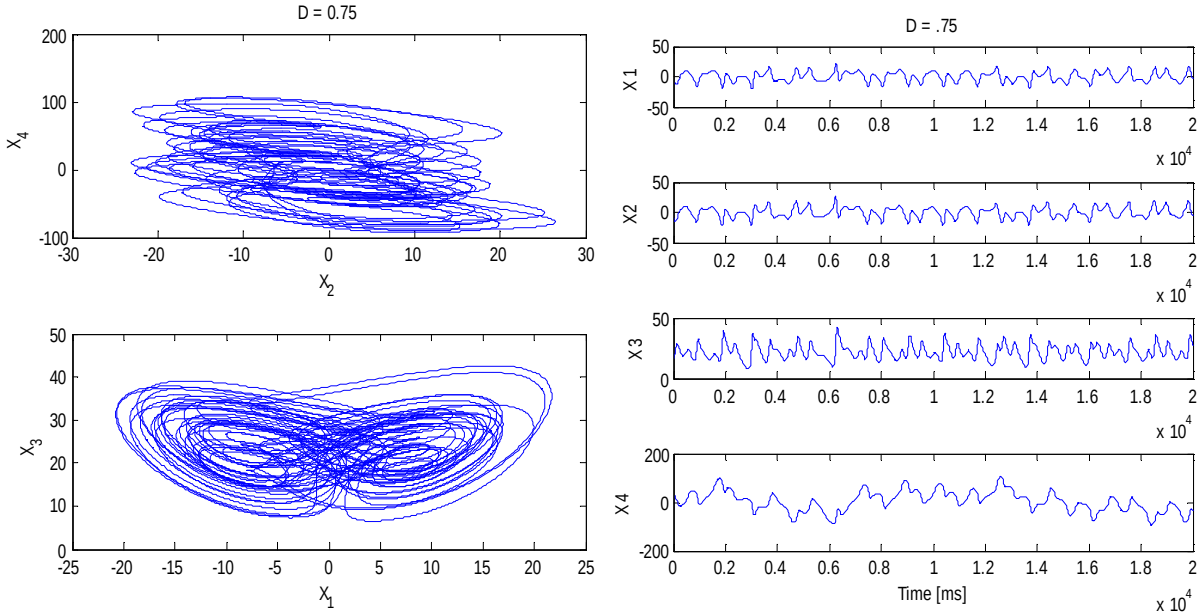


(a)

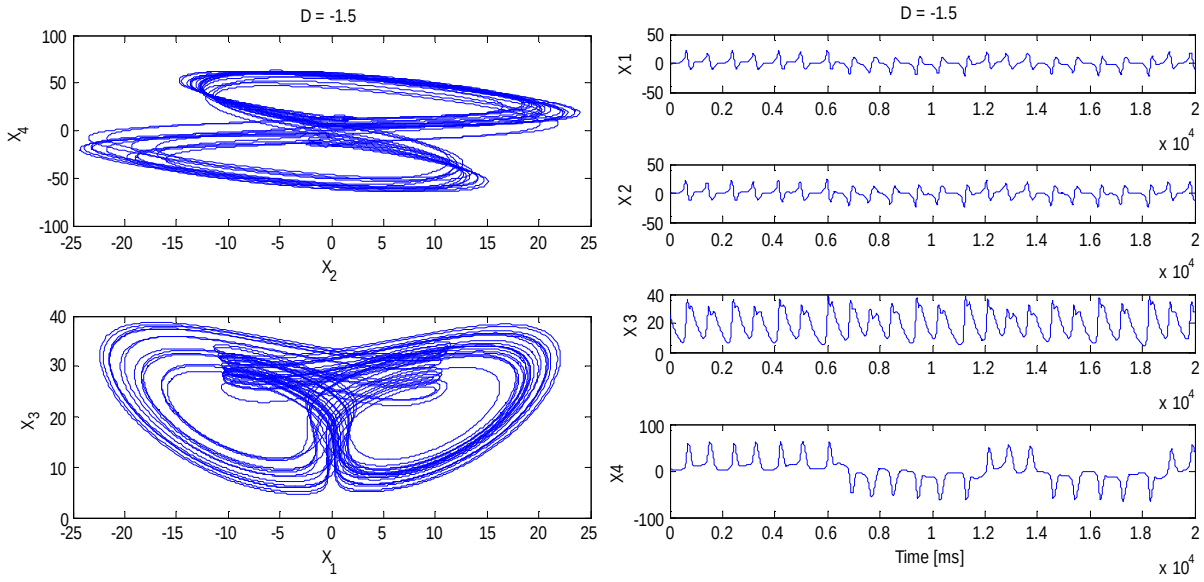


(b)

Figure 6.6 Lempel-Ziv complexity for a feedback-controlled system when A=36, B = 3, C = 20, $30 \leq D \leq 39$, ΔD = 0.01, n = 16300, ε = 0.001 (a) τ = 100 with maximum LZC=974 (b) when τ = 50 with maximum LZC =761.

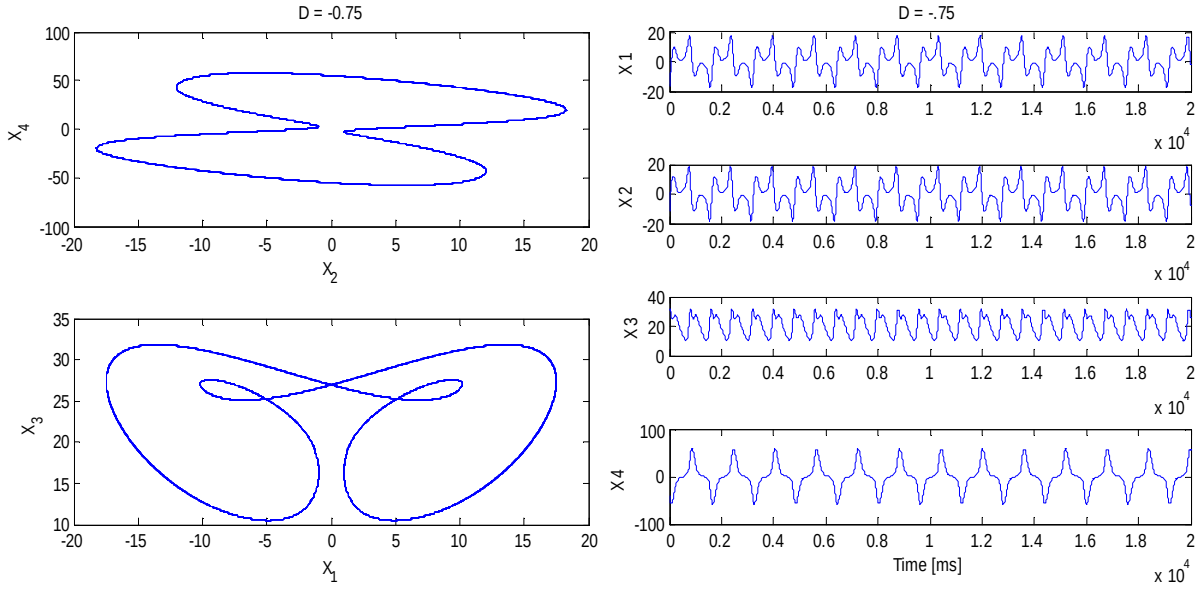


(a)



(b)

Figure 6.7 Feedback-controlled system phase portraits and time series (a) hyperchaotic state, D=0.75 (b) Chaotic state, D = −1.5.



(c)

Figure 6.7, cont. Feedback-controlled system phase portraits and time series (c) Periodic state, $D = -0.75$.

6.4 LZ Complexity for other Feedback-Controlled Systems

In this section, we study two other high-dimensional feedback-controlled systems with the LZC advance-time sampling method. Results are compared to the Lyapunov exponent spectrum. The immediate conclusion that arises from the comparison of the results confirms our conjecture about the LZ complexity of the hyperchaotic system. The first system is an eight dimensional feedback-controlled system defined by

$$\begin{aligned}
\frac{dx_1}{dt} &= A(x_2 - x_1) + x_4 \\
\frac{dx_2}{dt} &= -x_1 x_3 + C x_2 \\
\frac{dx_3}{dt} &= x_1 x_2 - B x_3 \\
\frac{dx_4}{dt} &= x_1 x_3 + D x_4 \\
\frac{dx_5}{dt} &= A(x_6 - x_5) + x_8 \\
\frac{dx_6}{dt} &= -x_5 x_7 + C x_6 \\
\frac{dx_7}{dt} &= x_5 x_6 - B x_7 \\
\frac{dx_8}{dt} &= x_5 x_7 + D x_8
\end{aligned} \tag{6.5}$$

And second system defined by,

$$\begin{aligned}
\frac{dx_1}{dt} &= A(x_2 - x_1) + x_4, \\
\frac{dx_2}{dt} &= -x_1 x_3 + C x_2, \\
\frac{dx_2}{dt} &= x_1 x_2 - B x_3, \\
\frac{dy_2}{dt} &= x_1 x_2 + D x_4.
\end{aligned} \tag{6.6}$$

where *A*, *B*, *C* are the constants and *D* is a control parameter. Hereafter, for simplification, the system in (Eqs. 6.6) is called the Din system. The dynamical behaviors of these systems, including Lyapunov exponents, LZC and phase portraits are explored. For the 8-dimensional feedback controlled system (Eqs. 6.5), the Lyapunov exponents spectrum for a specific values of $A = 30$, $B = 3$ and $C = 20$, when D varies from –1.5 to 1.5 with step 0.01 is shown in Figure 6.8. This diagram covers completely the real parameter region of D. Periodic behaviors are for $-1.14 \leq D \leq -0.85$. Chaotic attractors occur at several windows at: $-1.46 < D \leq -1.16$, $-0.84 < D \leq -$

0.75 and $-0.73 < D \leq 0.16$ and hyperchaotic responses for $0.16 \leq D \leq 1.37$. For $1.38 \leq D \leq 1.5$, there are three hyperchaotic attractors at $D = 1.39$, 1.40 and 1.45, while for other values of D, the system diverges, hence there is no bound solution. Figure 6.9 illustrates the phase portraits for the 8-dim Feedback controlled system when A=30, B=3, C=20. The examples of the hyperchaotic states occur at D =1 and 1.25; periodic states at $D = -0.75$, -0.22, -2.5 and an example of chaotic state at $D = -1.25$.

The corresponding LZC computation result is performed based on a similar decision rule applied to the 4-dim feedback controlled system. This is done by partitioning the projection of phase space on x_1-x_3 space into four zones: ($x_1 \geq 0$; $x_3 \geq 25$), ($x_1 \leq 0$; $x_3 \geq 25$), ($x_1 \leq 0$; $x_3 \leq 25$), ($x_1 \geq 0$; $x_3 \leq 25$). After removal of $5\text{x}10^4$ transient responses, if the trajectory lands in zone 1 and 3, bit 1 is inserted into the sequence with length n=16700, and bit 0 otherwise. The integration with step size 0.01 is performed for a specific values of the control parameters set at $A = 36$, $B = 3$ and $C = 20$, when D is varied from –1.5 to 1.2 with step 0.01. Advance time-step is equal to 10. LZC plot is shown in Figure 6.10. It is observed that LZ complexity has a non-decreasing characteristic envelop behavior in the hyperchaotic region similar to the previous 4-dim feedback system. In chaotic regions, the LZC measure increases from low values to high values and again drops in periodic windows, but keeps increasing when parameter D increases from 0.16 to 1.5. We will see that this characteristic signature does not appear in the LZ complexity diagram of the Din system.

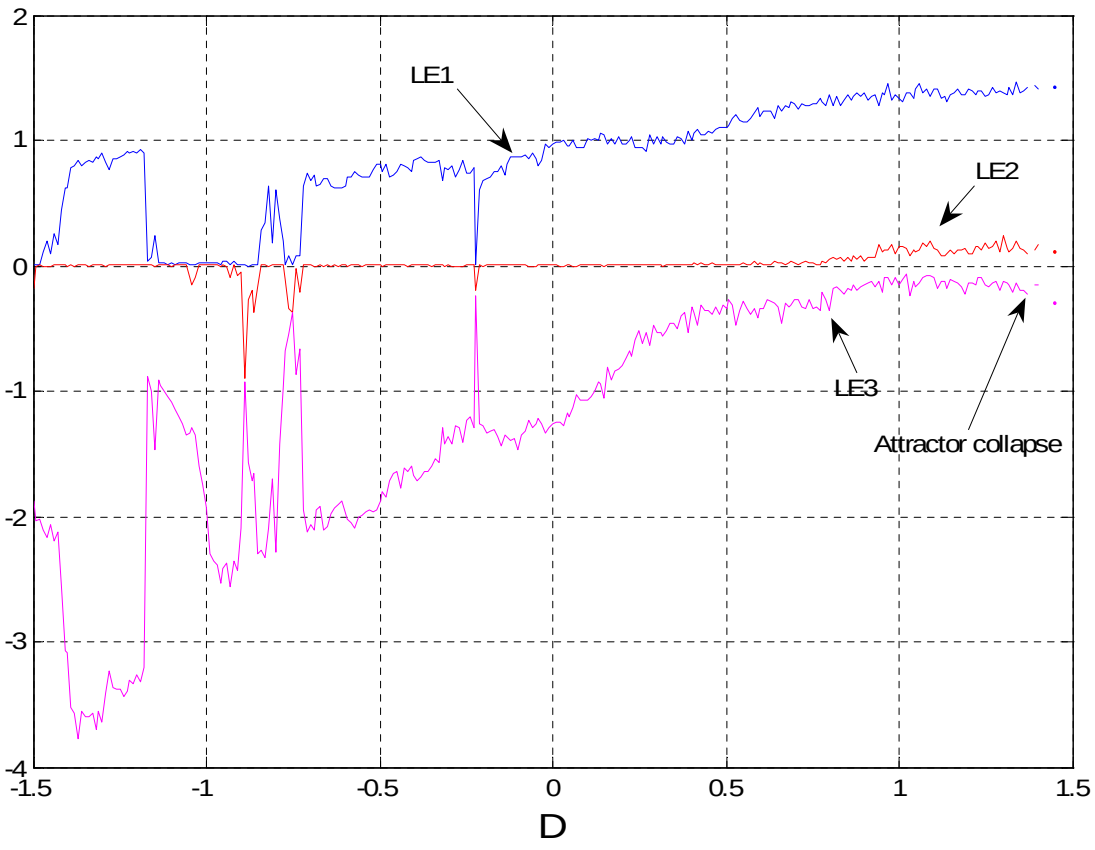


Figure 6.8 Lyapunov exponents $\lambda_1, \lambda_2, \lambda_3$ for 8-dim feedback-controlled system for $-1.5 \leq D \leq 1.5$, A=30, B=3, C=20, over 10^5 iterations with 10^4 transients removal, integration time-step 0.01, GSR step 1.

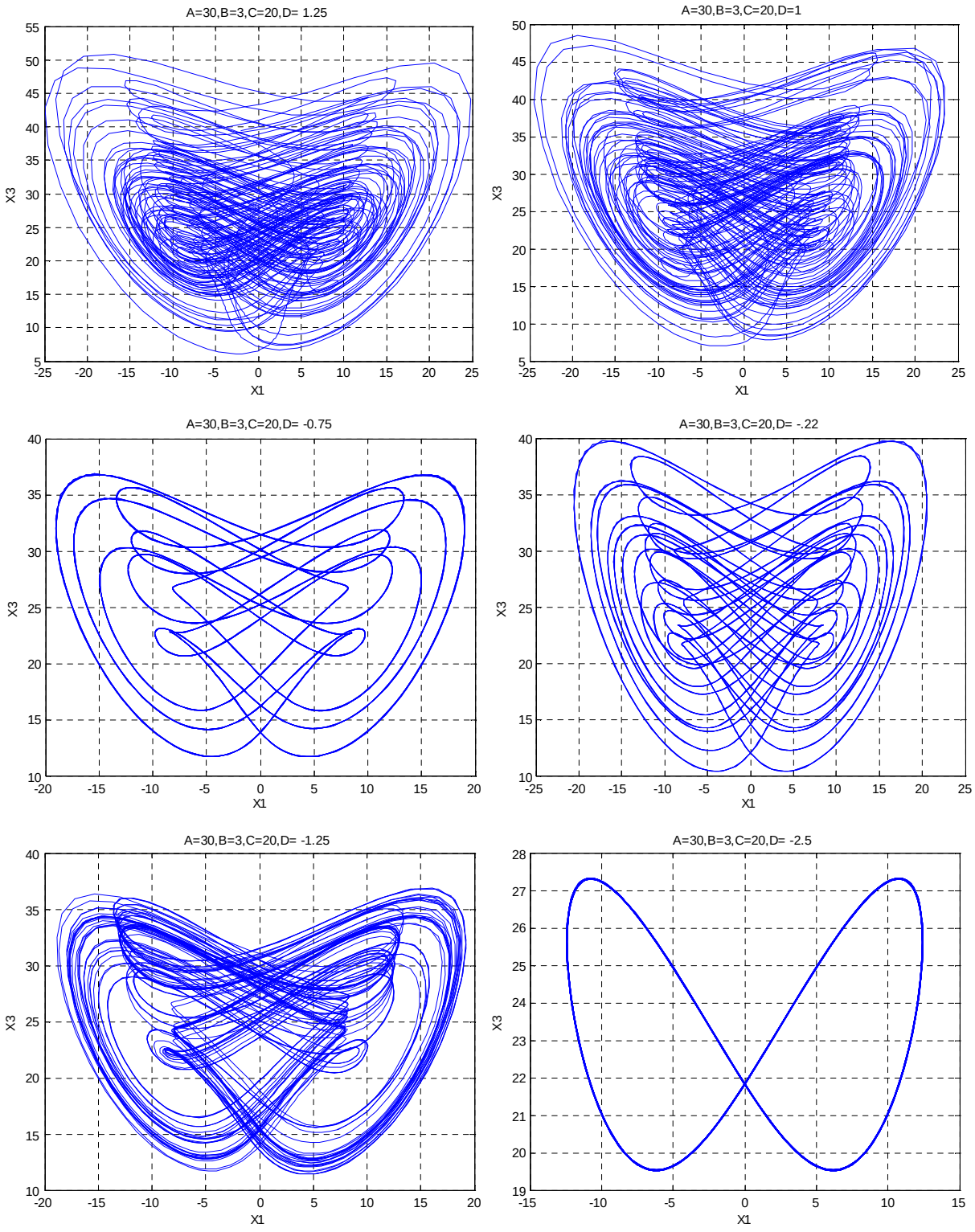


Figure 6.9 8-dim feedback-controlled system phase portraits. Hyperchaotic state, D=1, 1.25 periodic states, $D = -0.75, -0.22, -2.5$; and an example of chaotic state at $D = -1.25$; A=30, B=3, C=20.

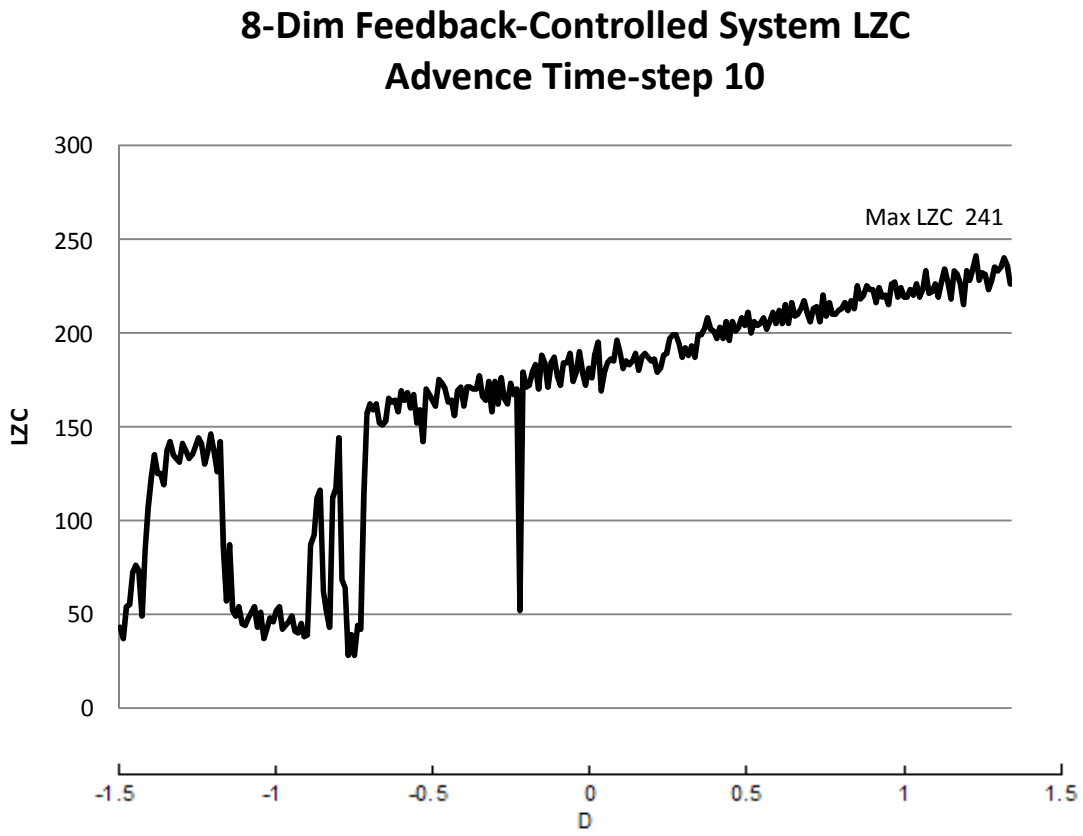


Figure 6.10 Lempel-Ziv complexity for 8-dim feedback-controlled system when A=30, B=3, C=20, D: $-1.5 \le D \le 1.5$ [ΔD=0.01], buffer size 16300, integration step-size 0.001 over 10^5, after 10^4 transients removed, advance time-step τ=10.

For Din system (Eqs. 6.6), the Lyapunov exponent spectrum for a specific values of A=16, B=3, C=10, when D varies from –20 to −2 with step 0.1 is shown in Figure 6.11. Periodic behaviors occur for $-13.5 \le D \le -12.6$ and $-2.9 \le D \le -2$. In addition, many other periodic regions occur at several windows when D is equal to −15.1, −10.9, −8.7; $-4.4 \le D \le -4.5$, −3.5 and −3.2. Chaotic responses occur in the $-20 \le D \le -13.6$ and $-12.6 \le D \le -3$ parameter ranges. There are no hyperchaotic attractors for $-20 \le D \le -2$. Figure 6.12 shows the Din system phase portraits in x_3-x_1 plane when the parameters are A=16, B=3 and C=10. Chaotic states occur at D =

−20, −6 and periodic states at $D = -13$ and −2.5. Figure 6.13 shows a typical chaotic state phase portraits in x_i-x_j plane when $D = -47.4$.

In Figure 6.14 we plot the LZC vs. D in the same parameter range. We observe that the LZ complexity values coincide with the above range for chaos regions and periodic windows. We notice that this system does not manifest the non-decreasing characteristic envelope behavior as expected from a hyperchaotic region. In chaotic regions, the LZC measure increases from low values to high values and again drops in periodic windows without any attractor collapse or diverging dynamics. Hence, in this range of the control parameter, hyperchaotic responses are not expected. Consequently, this is evidence that the LZC measure can also distinguish chaos from hyperchaos. These characteristic signatures are powerful indicators in classifying dynamical systems behavior.

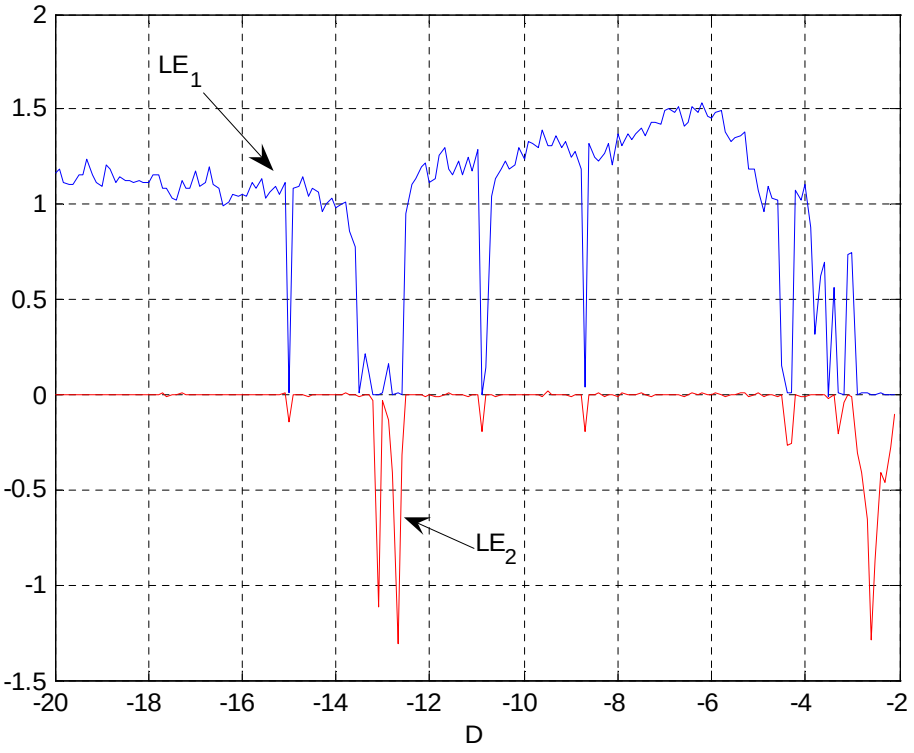


Figure 6.11 Din system Lyapunov exponents λ_1, λ_2 for D: $-20 \leq D \leq -2$, [ΔD=0.1], A=16, B=3, C=10, 5×10^5 iterations, 10^5 transients, ε =0.001, GSR step 1.

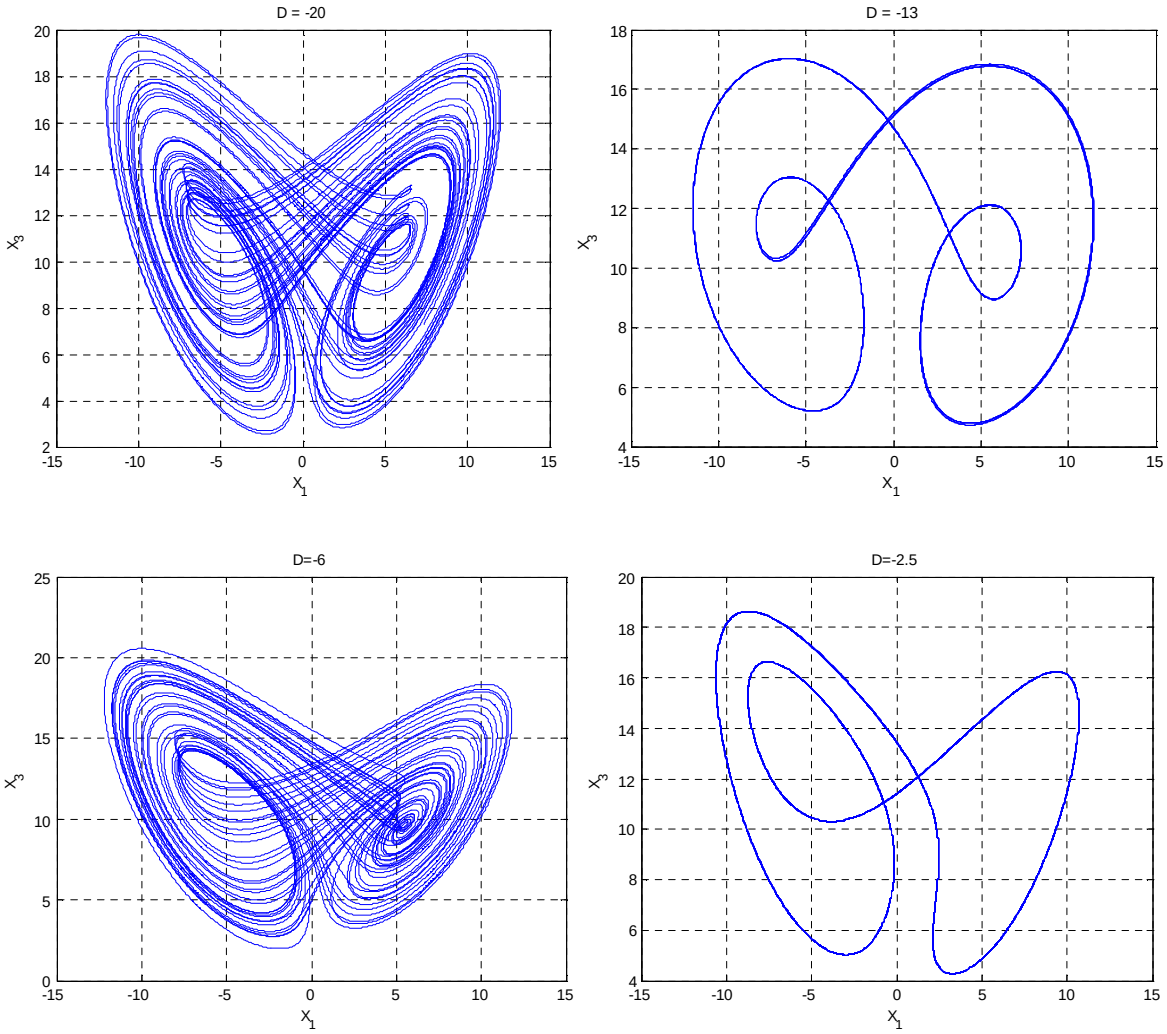


Figure 6.12 Din system phase portraits. Chaotic states occur at $D = -20, -6$ and periodic states at $D = -13$ and -2.5.

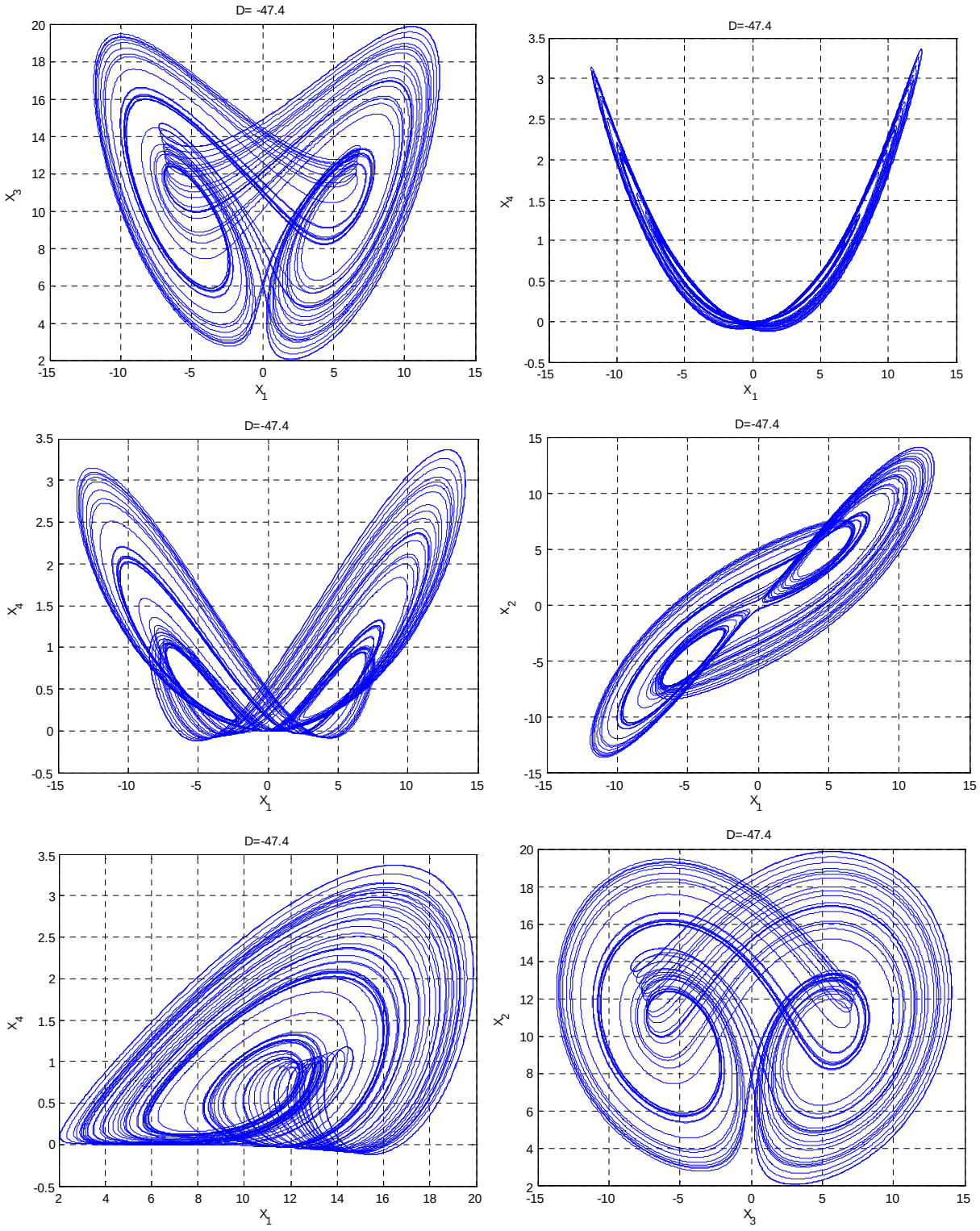


Figure 6.13 Din system typical chaotic state phase portraits in x_i-x_j plane when $D = -47.4$. A=16, B=3, C=10.

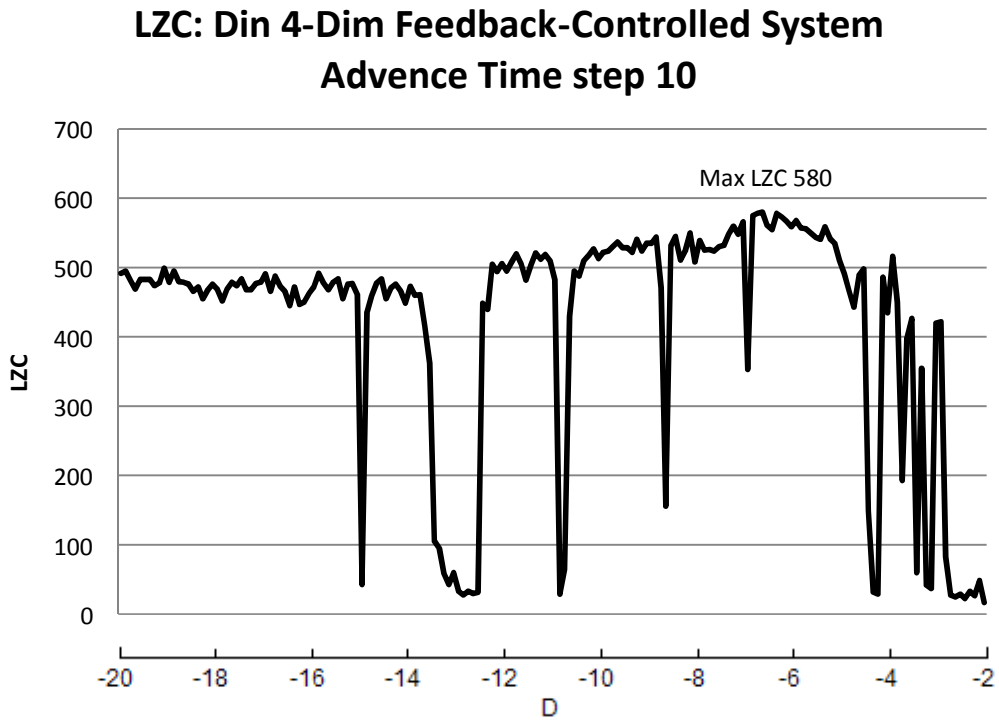


Figure 6.14 LZ complexity for Din system when $A = 16$, $B = 3$, $C = 10$, $-20 \leq D \leq -2$, $\Delta D = 0.1$, buffer size 16300, integration step-size 0.001 over 10^5, 10^4 transients, advance sampling steps $\tau = 100$.

6.5 Conclusion

In chapter 6 we presented an overview of complexity measures in hyperchaotic systems information processing. The advance sampling method is designed to overcome the scarcity of phase space samples in algorithmic complexity measure estimation and implemented algorithmically on several feedback controlled systems. Special attention is paid to the complexity of information for an entire region of chaos-hyperchaos transition. We find that LZC values for 4-dim feedback controlled hyperchaotic system have a non-decreasing characteristic behavior just before the occurrence of attractor collapse when a diverging dynamics occurs. On the other hand, in chaotic regions the LZC measure increases from low values to high, and then

drops in periodic windows, but does not a sustain non-decreasing envelope. We report this phenomenon as a characteristic mechanism of chaotic to hyperchaotic state transition. These characteristic signatures are powerful indicators in classifying dynamical systems behavior.

CHAPTER 7

SYNCHRONIZATION IN COUPLED CHAOTIC OSCILLATORS

7.1 Introduction

The adjustment of rhythms due to an interaction is known as synchronization. This phenomenon is an example of coherence between initially irregular oscillations of coupled self-sustained oscillatory systems. Modern technology has many examples of useful applications of synchronization. For instance, a coherent laser beam is the combined photons of trillions of atoms synchronized to emit in phase or SQUIDs rely on the synchronization of the quantum states of two Josephson junctions, and there is the phenomenon of synchronization in chaotic systems which has potential importance in secure communications. The last application is because of the noise like appearance of the time series generated by the chaotic systems, and also because many chaotic systems are easy to realize, so they are widely used in the real-time environments. For example, these circuit-based systems, which often operate at low power, are inexpensive to produce and are useful in applications in the consumer electronics arena. [100−108].

In this chapter we investigate the dynamics of non-identical nonlinear oscillators. These systems are coupled autonomous Van der Pol (VP) and Duffing (DF) oscillators. A Van der Pol oscillator has a dissipation term, which is negative for small amplitudes, supplying energy to the system, and positive for large amplitudes. Therefore, spontaneous self-sustained oscillations can be observed even without periodic driving. This system is the result of Van der Pol and Van der Mark studies of a simple electronic circuit for which the negative dissipation was provided by a negative resistance region in the I-V characteristic of a neon tube. When this oscillator is driven

at a frequency which is different from its free oscillation frequency, the nonlinear phenomenon of frequency locking is observed. Another well-known nonlinear oscillator used to model various physical, engineering and even biological problems is the Duffing oscillator which provides a demonstration of several interesting phenomena such as nonlinear resonance, bifurcations, and hysteresis. The system equation of the forced vibration of a nonlinear system with viscous resistance and a cubic quasi-elastic characteristic is of the form:

$$\ddot{x} + \mu\dot{x} + \omega_0^2 x + \beta x^3 = F\sin(\Omega t). \tag{7.1}$$

The amplitude of vibration (A) can be determined approximately from the equation:

$$\left(\Omega^2 - \omega_0^2\right)A - \beta A^3 \pm F\sqrt{1 - \frac{\mu^2}{F^2}(A\Omega)^2} = 0. \tag{7.2}$$

For $\beta > 0$, if the amplitude of the forcing function is held constant, there exists a range of forcing frequencies for which three possible output amplitudes are possible. Then the function $A(\Omega)$ takes the form shown in Figure 7.1. This is the classical resonance curve when the response is periodic with the same period as the driving force. Unstable solutions are indicated by a dashed line. A hysteretic effect occurs for increasing and decreasing frequencies. This is known as the jump effect and can be observed in electrical and mechanical systems.

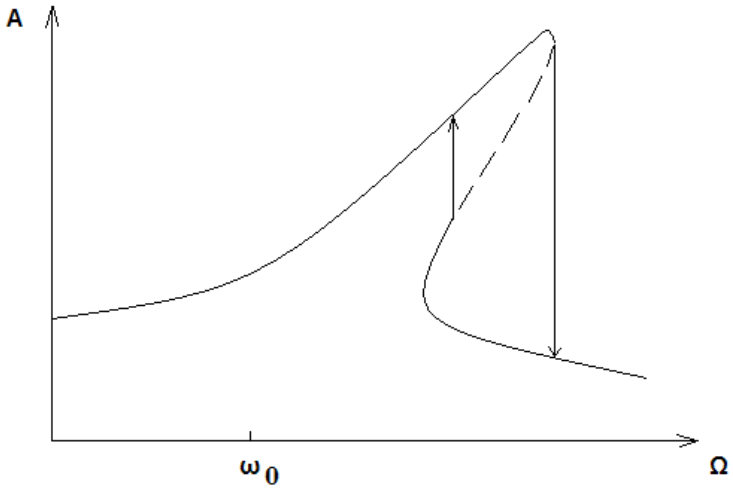


Figure 7.1 Duffing system resonance curve $A(\Omega)$

The driven Duffing oscillator also shows chaotic behavior. To illustrate this we plot the bifurcation diagram (Figure 7.2), Lempel-Ziv complexity measure (Figure 7.3), and topological entropy (Figure 7.4) for a certain range of its control parameters. These are set to $\beta = 1$, $\Omega = 1$, $\mu = 0.5$, and F increasing from 0.35 to 0.663, with increment 0.001. τ is the normalized time and is equal to $\omega_0 t$. For topological computation, phase space region of attractors is locally partitioned and subdivided into 40 by 40 units of the total 1600 cells. Then the number of certain final trajectories is determined in a given region of phase space for a specific parameter. From this quantity, one can compute the probability that certain trajectories will be in a given region of phase space. From the probability density, it is easy to find the entropy for each driver amplitude F in the range.

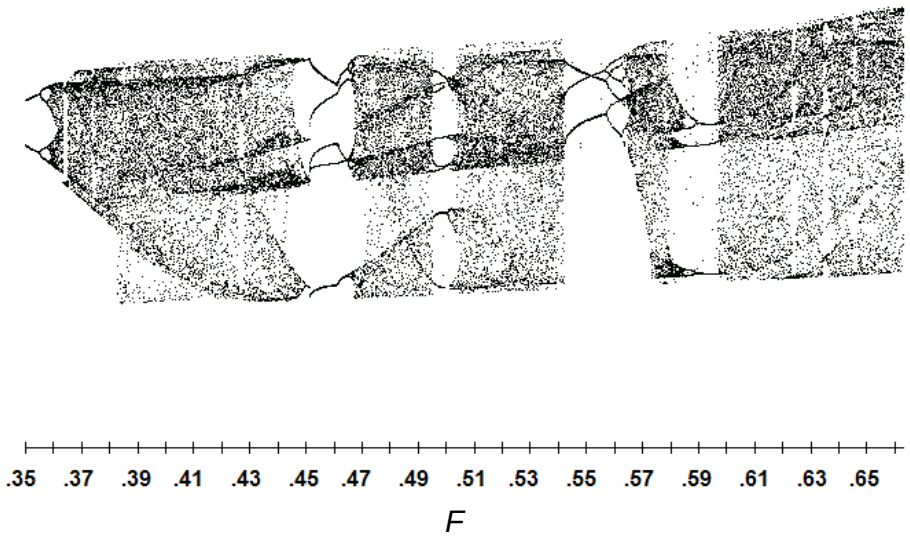


Figure 7.2 Duffing system bifurcation diagram $0.35 \leq F \leq 0.663$, ΔF=0.001, Ω=1, μ=0.5.

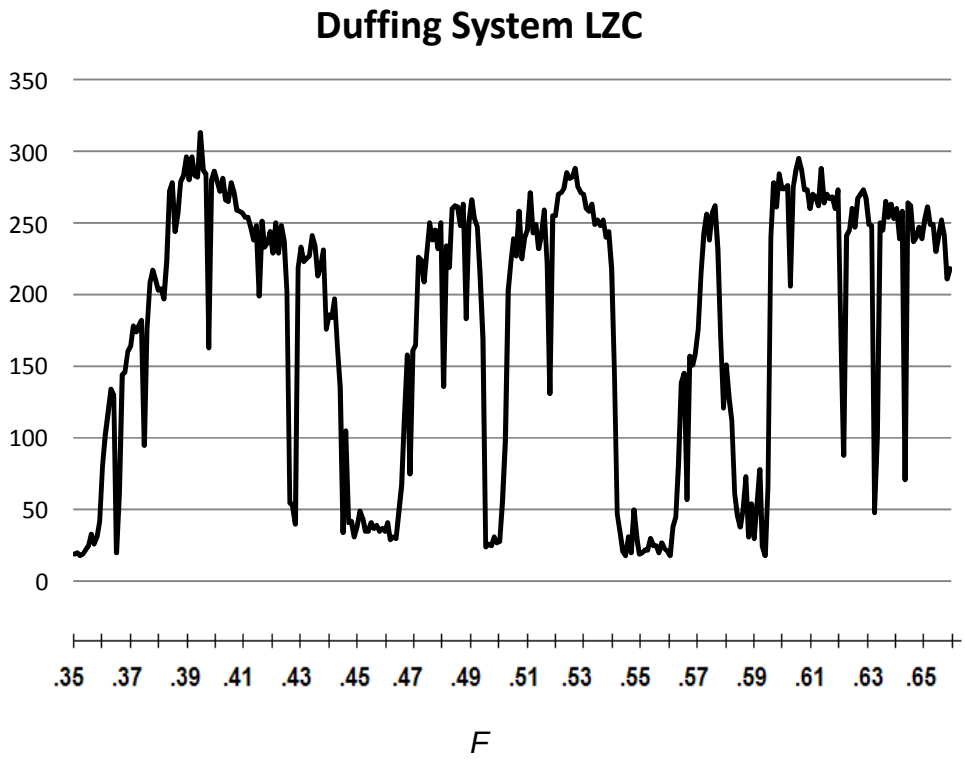


Figure 7.3 Duffing system LZC plot for $0.35 \leq F \leq 0.663$, ΔF=0.001, Ω=1, μ=0.5.

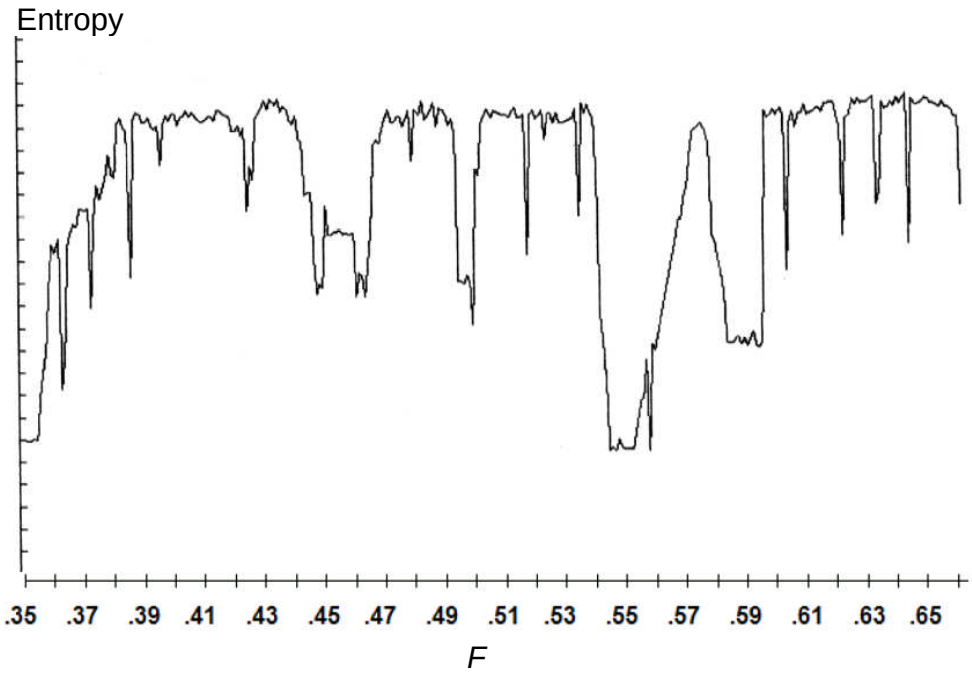


Figure 7.4 Duffing system topological entropy plot, $0.35 \leq F \leq 0.663$, ΔF=0.001, Ω=1, and μ=0.5, n=3×10^5, tolerance=0.005.

7.2 Coupled Non-Homogeneous Chaotic System

A non-homogeneous coupled system has at least two different oscillators. This generates very rich dynamic phases for the various choices of the system parameters and the coupling constant. To examine the transition between dynamic phases, we construct the bifurcation diagram and the phase diagram of the parameter space from the Lyapunov exponent spectrum, for a specific coupling constant. As the coupling constant gets larger, the dynamics of the system becomes simpler. The synchronizations of the system are also investigated by varying the system parameters. The model equations are defined by the following dimensionless systems:

$$
\begin{aligned}
\frac{dX_1}{dt} &= Y_1, \\
\frac{dY_1}{dt} &= -M\left(1-X_1^2\right)Y_1 - X_1 + K\left(X_2 - X_1\right), \\
\frac{dX_2}{dt} &= Y_2, \\
\frac{dY_2}{dt} &= -AY_2 + X_2 - X_2^3 + K\left(X_1 - X_2\right).
\end{aligned}
\tag{7.3}
$$

$$
J = \begin{pmatrix}
0 & 1 & 0 & 0 \\
-2MX_1Y_1 - K - 1 & M(1-X_1^2) & K & 0 \\
0 & 0 & 0 & 1 \\
K & 0 & 1-3X_2^2 - K & -A
\end{pmatrix}.
\tag{7.4}
$$

$$
\nabla.\vec{F} = \sum_{i=1}^{4} \lambda_i = M\left(1-X_1^{\,2}\right) - A \tag{7.5}
$$

Coupling constant $K = 0$, (Eq. 7.3) describes the uncoupled oscillators. The coupling between the two oscillators is set in a symmetrical way that depends on the difference of the system variables. *A, M* and *K* are the control parameters for the coupled system. In the case of *K* equal to zero, the coupled system may be considered as a single, damped DF oscillator driven by a VP oscillator. There is a significant difference between a damped DF oscillator driven by an external force and the coupled system under study. A driven DF oscillator is affected by the external force from one side, but the two oscillators of the coupled system affect each other from both sides. To determine the complete dependency of the system on the control parameters *M* and *A*, the Lyapunov exponents spectrum has been determined. This is done by isolating the regions with positive maximum Lyapunov exponents. We use the initial conditions: $X_1 = 0.8$, $X_2 = 2$, $Y_1 = 0$ and $Y_2 = 0$ and a fixed coupling constant $K = 0.5$ for the third control parameter. To confirm the

correctness of simulation results, Lyapunov exponent spectrum addition rule implemented and tested against the divergence of flow. Figure 7.5 illustrates the proximity of these two quantities.

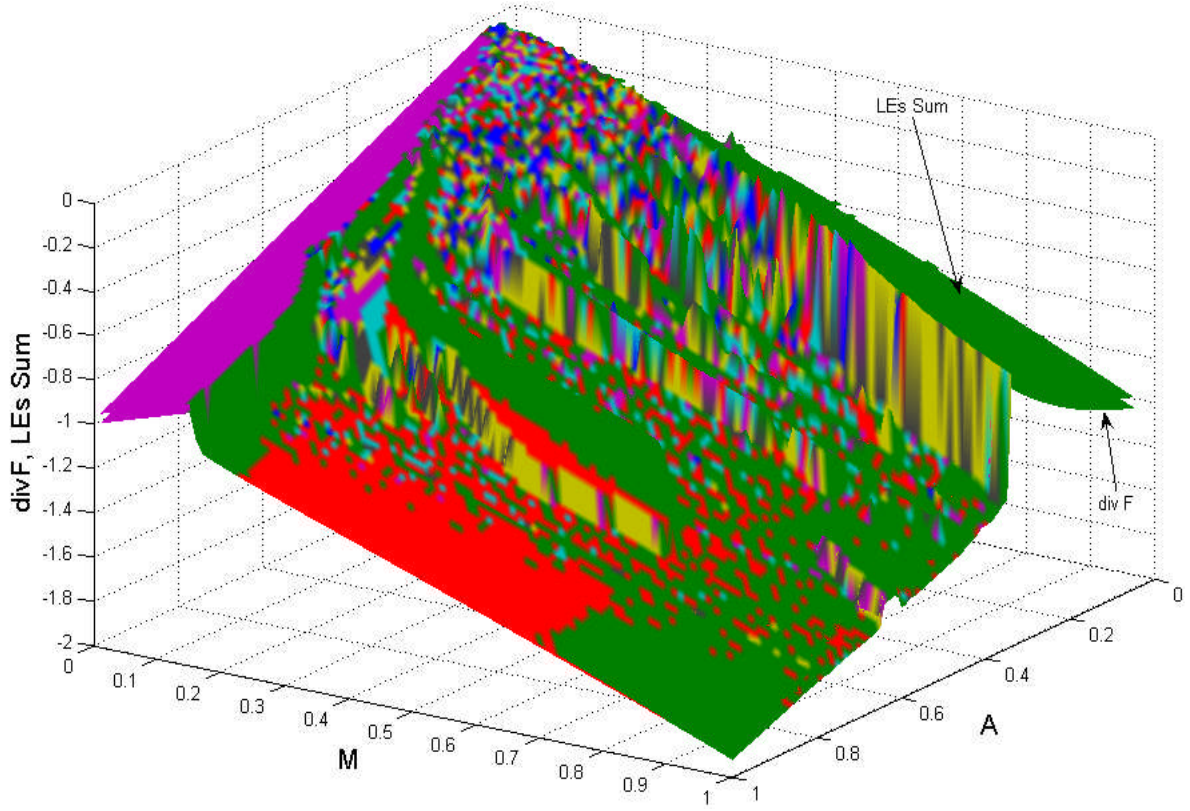
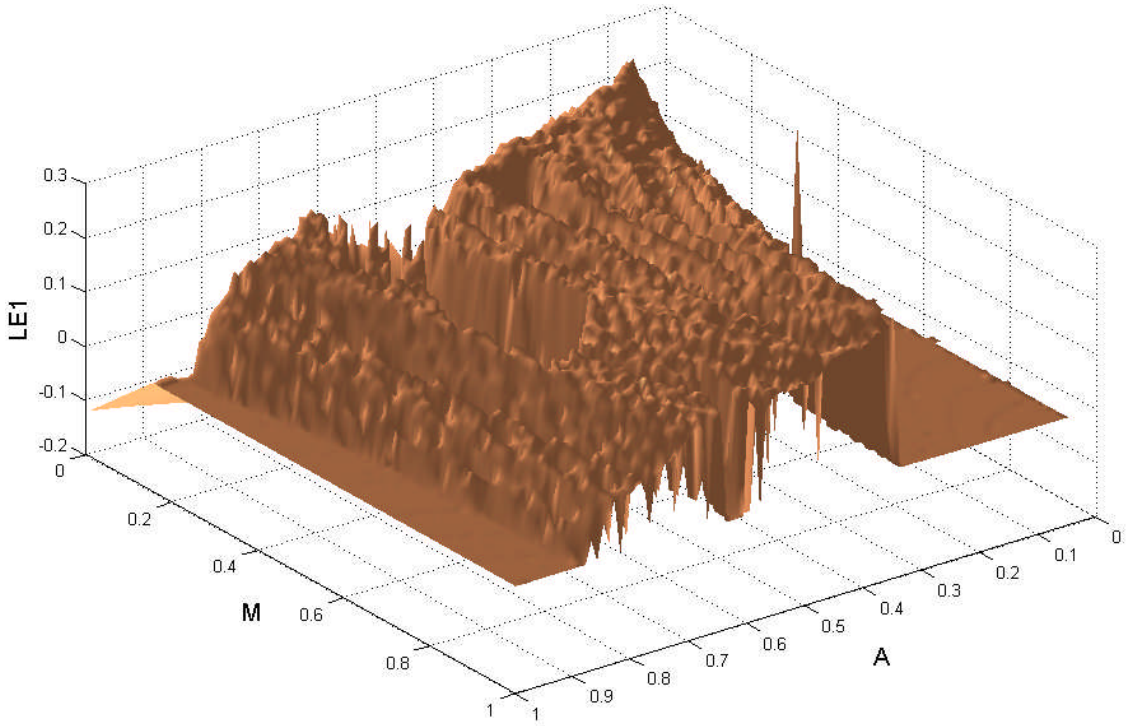
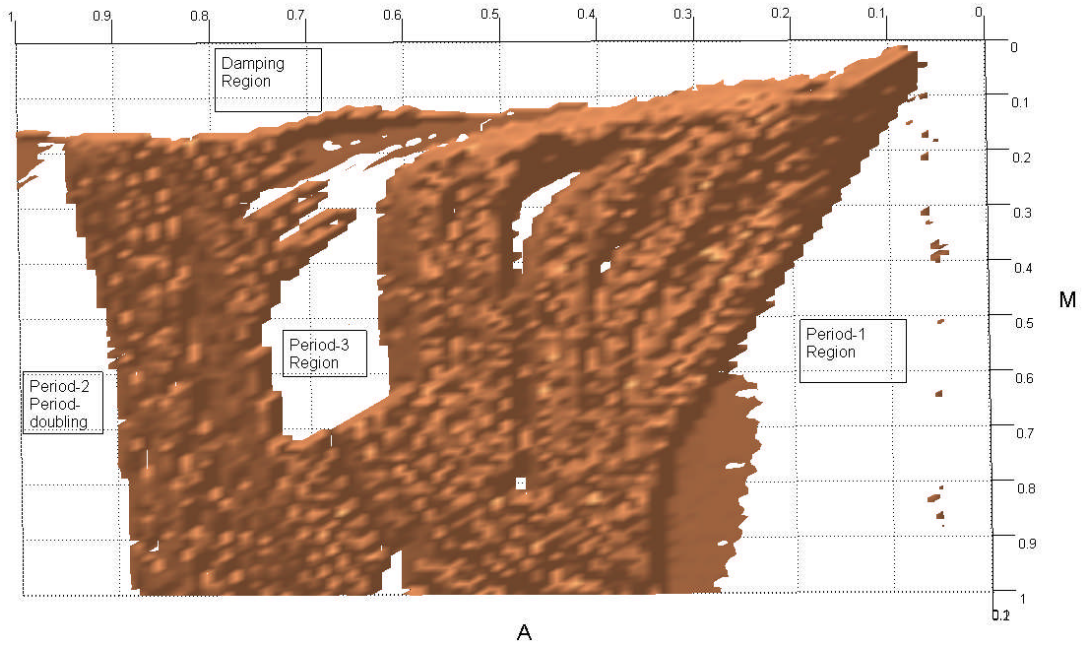


Figure 7.5 Addition rule verification. $0 \leq M \leq 1$ [ΔM = 0.01], $0 \leq A \leq 1$, [ΔA=0.01], K=0.5, 2×10^5 iterations, time-step 0.1, 10^4 transients, GSR step 1.

The resulting phase diagram is shown in Figure 7.6. The phase diagram is drawn in the range from 0 to 1 for the parameters *M* and *A*. The region of phase is subdivided into 100 by100 cells. Thus, our phase diagram is digitized by unit of 0.01 for the *M* and *A* axes. We construct two phase diagrams of the parameter space by showing the positive values of the first and second Lyapunov exponents. The latter is in Figure 7.7. The dynamical transitions from periodic to chaotic states is traced by varying *M* and *A*. Also by exploring the positive second Lyapunov exponent, small regions of hyperchaotic responses are identified.



(a)

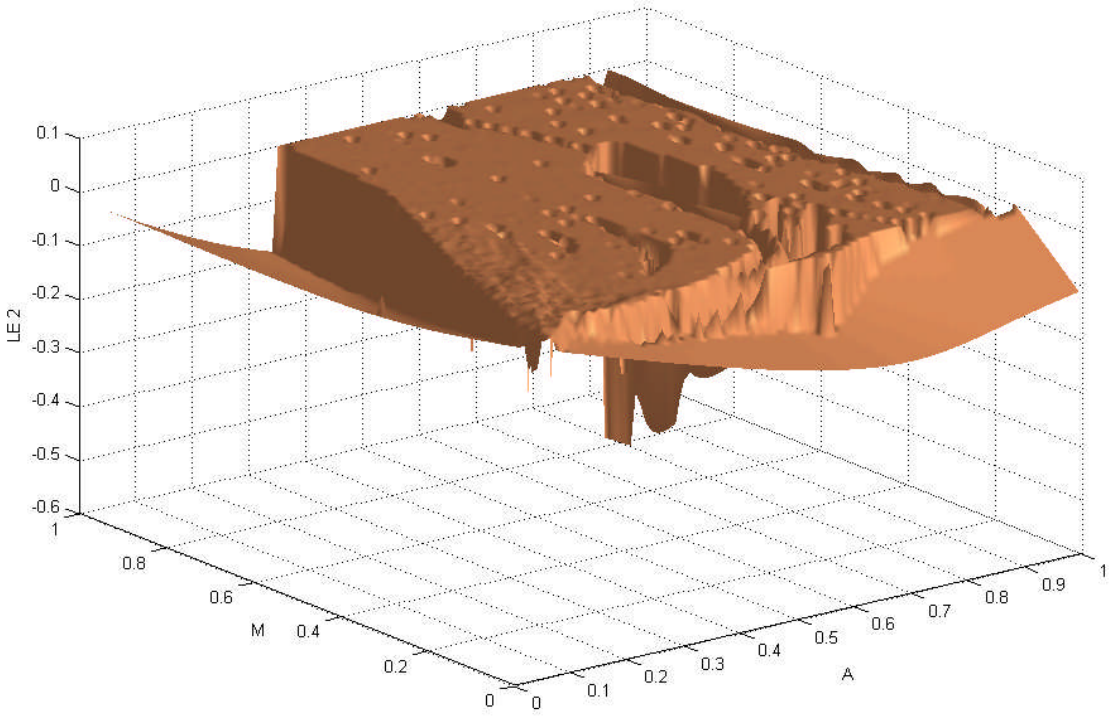


(b)

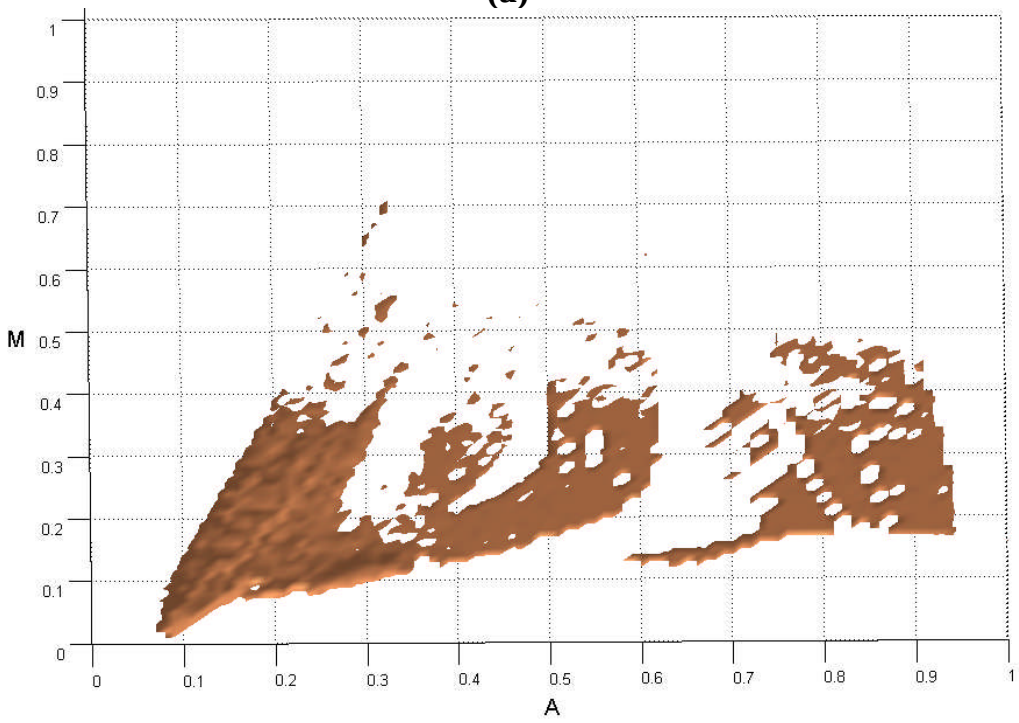
Figure 7.6 (a) First Lyapunov exponent (b) Positive first LE. $0 \leq M \leq 1$, [ΔM=0.01], $0 \leq A \leq 1$, [ΔA=0.01], K=0.5, over 2×10^5 iterations, integration time-step 0.1 and after 10^4 transients, GSR step 1.

The system shows very rich dynamic phases for various values of the parameters *M* and *A*. There are regions of periodic states and a periodic doubling state and large area of chaotic states. The results of the phase diagram can be interpreted by using the following regions of dynamic states. In the period-1 region, as the forcing parameter *M* increases further in the region of small values of *A*, the period-1 region expands. The region limit reaches $A = 0.33$ at $M = 1$. This is an expected behavior because as the parameter *M* increases the VP oscillator's limit-cycle attractor becomes stronger. There are many isles of period states that are mostly orbits of period-*m*, where *m* is greater than 3. In period-2 regions, the middle of the chaotic region, there is an island of a period-2 state and a long region on the right side of the phase diagram which starts from $A \sim 0.9$. The regions develop from the chaotic region through periodic doubling states. In period-3 region, a big peninsula of a period-3 state develops in the chaotic sea from the damping region at the bottom of the phase diagram. We find a period-1 region between the damping region and the period-2 region around $A = 1$. The dynamics of this region shows again the motion of the limit-cycle through a supercritical Hopf bifurcation. This bifurcation begins at $M = 0.19$ and $A = 0.88$. If the parameter *A* is increased to a value over 1, it is found that the period-1 phase occurs again for all values of *M* (from 0 to 1). By increasing the damping parameter *A* in the region of small values of *M*, the damping phases appear as a point attractor of the DF oscillator. The region gets wider as the damping coefficient increases up to 1. This is the summary of results for a phase diagram in (*M* and *A*)-parameter space for the coupling constant $K = 0.5$. Next, we determine the dependences of the characteristic properties of the dynamics on the two system parameters *M* and *A*, to understand the roles of each system parameter in the dynamics of the coupled system. First we study the dominant role of each parameter for each attractor, and the dynamics of the region at $A > 1$ or $M > 1$ are also investigated by using two

different approaches. To find the roles of two attractors from the dynamics of this coupled system is not a simple task. To do so, in one system we vary the value of A with M= 0.01. For the other, we vary the value of M with A= 0.01. In the former approach, that is, in the case of weak dissipation of the DF oscillator, we find periodic states after fairly long chaotic transients. We also find that the chaotic transient states occur intermittently and that there are not only period-1 states but also multi-period states. It is found that the chaotic transients get longer as M is increased further. In the latter approach, we find that the period-1 limit-cycle states appear again as A increases for fixed M = 0.01 and K= 0.5. For the same initial condition as above, the period-1 state begins to occur at A=18.16, but the system very slowly approaches a stable period-1 state. As stated above, the coupled system has undergone a supercritical Hopf bifurcation. In fact, these dynamics are wholly unexpected since it is naturally expected that the dynamics of a coupled system should be subject to the point attractor of a Duffing oscillator which has a much more dominant system parameter than the other has (in this case, $A>>M$). The dominance of an attractor, based on its relative strength, clearly appears for each parameter at a certain level, as shown in the regions near the x-axis and the y-axis in Figure 7.6 (b). However, as stated above, the dominance is not so perfect and simple because there are some defects in the region dominated by an attractor. Hyperchaotic regions are shown in Figure 7.7 (b). This region contains attractors of the limit cycle of a VP oscillator which is transformed to a kind of strange hyperchaotic attractor by a coupled DF oscillator. Figure 7.8 shows a typical hyperchaotic attractor at K = 0.5, M = 0.29, A = 0.22. The corresponding Lyapunov exponents, λ_1 and λ_2, are 0.21 and 0.014 respectively. Figure 7.9 shows the third and fourth LE. In addition the Kaplan-York fractal dimension is illustrated in Figure 7.10.



(a)



(b)

Figure 7.7 (a) Second Lyapunov exponent (b) Positive second LE, hyperchaos regions. $0 \leq M \leq 1$, [ΔM=0.01], $0 \leq A \leq 1$, [ΔA=0.01], K=0.5, over 2×10^5 iterations, integration time-step 0.1, transients 10^4.

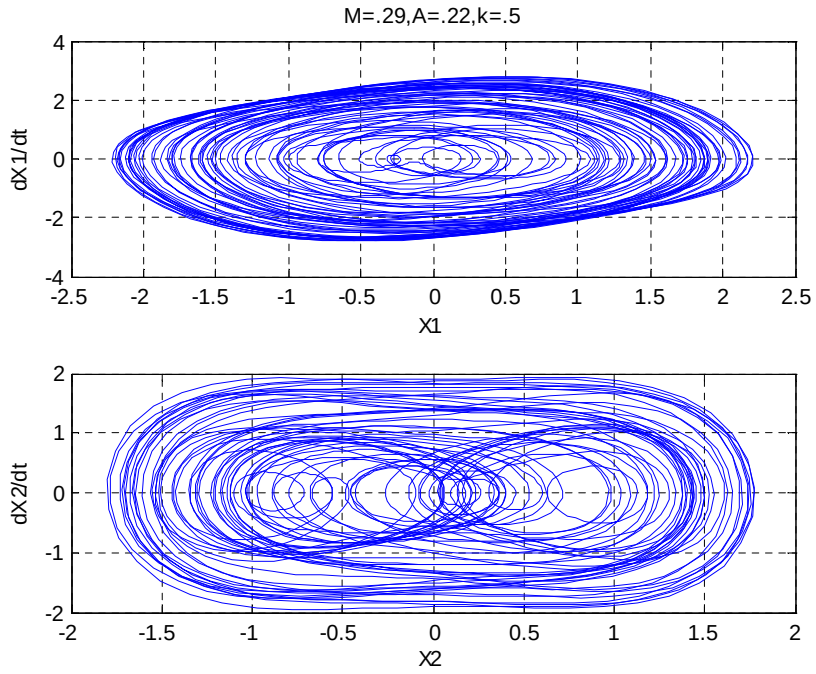
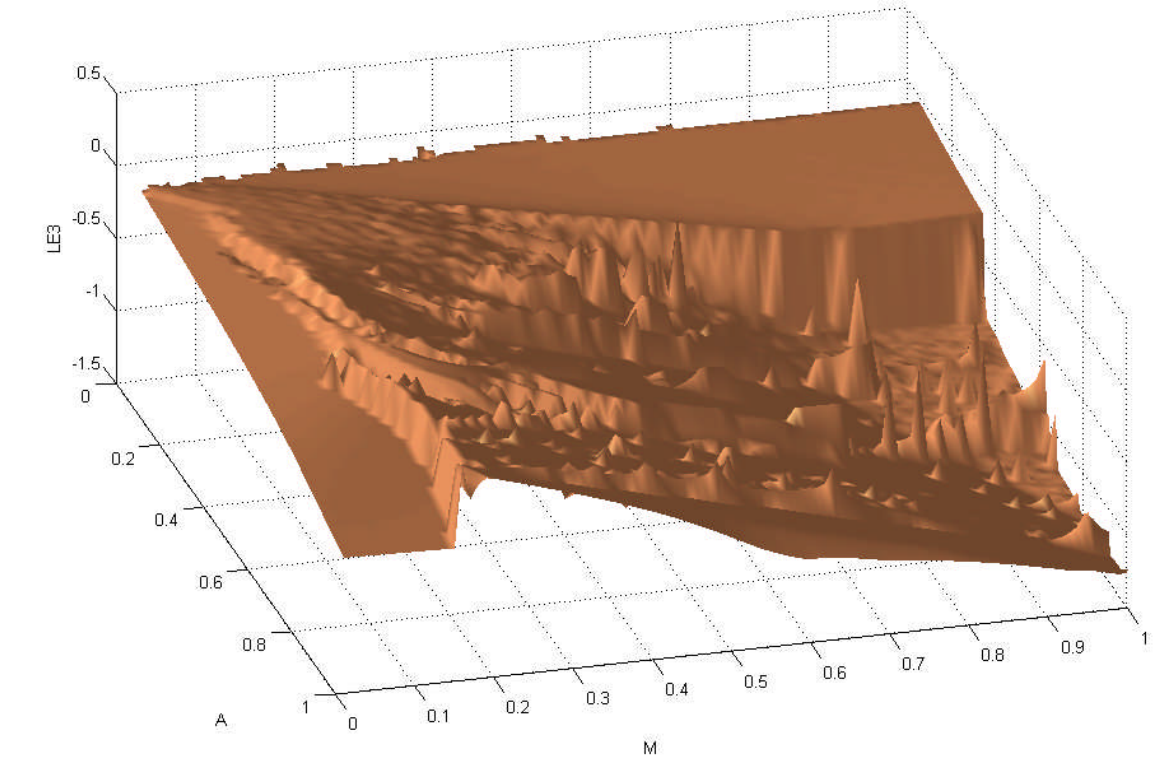
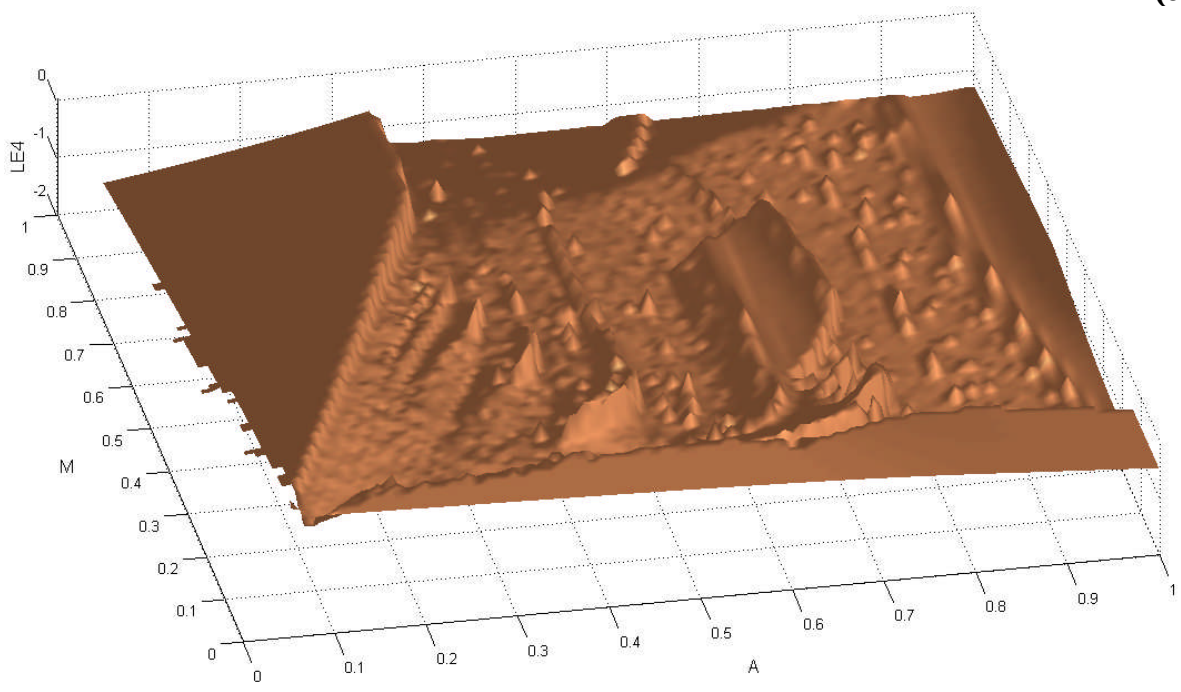


Figure 7.8 Hyperchaotic attractor at $K = 0.5$, $M = 0.29$, $A = 0.22$. LE1 = 0.21, LE2 = 0.014.



(a)



(b)

Figure 7.9 (a) Third LE (b) Fourth LE. Control parameters: $0 \leq M \leq 1$, [ΔM=0.01], $0 \leq A \leq 1$, [ΔA=0.01], K=0.5, 2×10^5 iterations, integration time-step 0.1, after 10^4 transients.

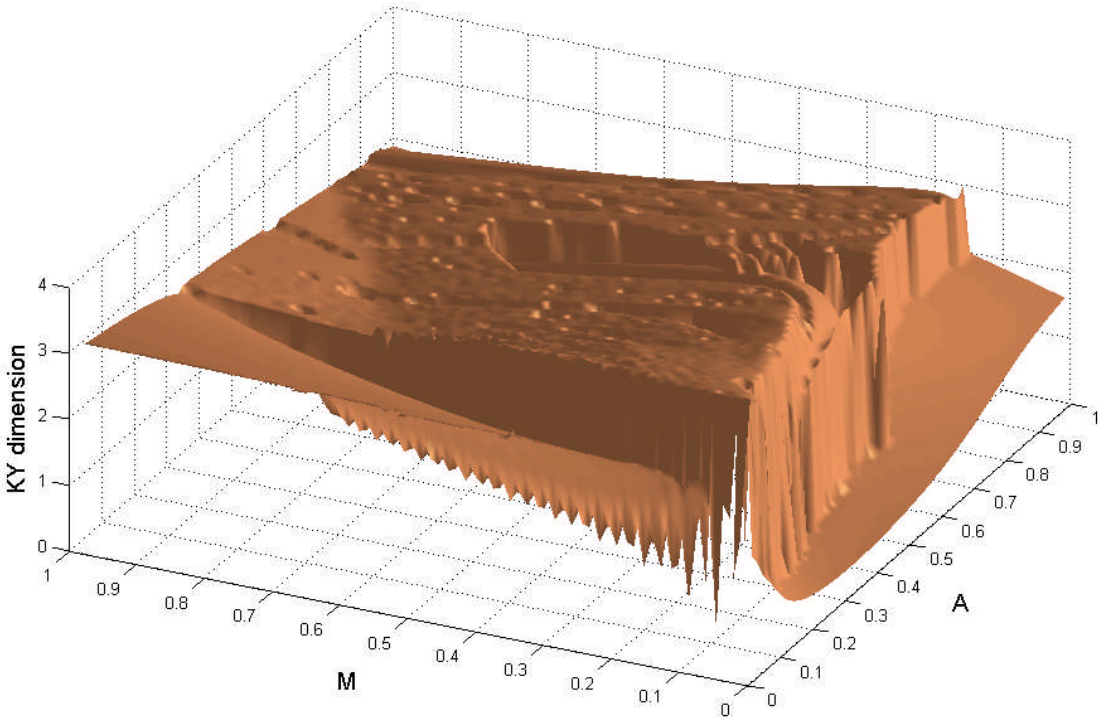


Figure 7.10 Kaplan-York fractal dimension, control parameters $0 \le M \le 1$ [ΔM=0.01], $0 \le A \le 1$, [ΔA=0.01], K=0.5, over 2×10^5 iterations with integration time-step 0.1 after 10^4 transients.

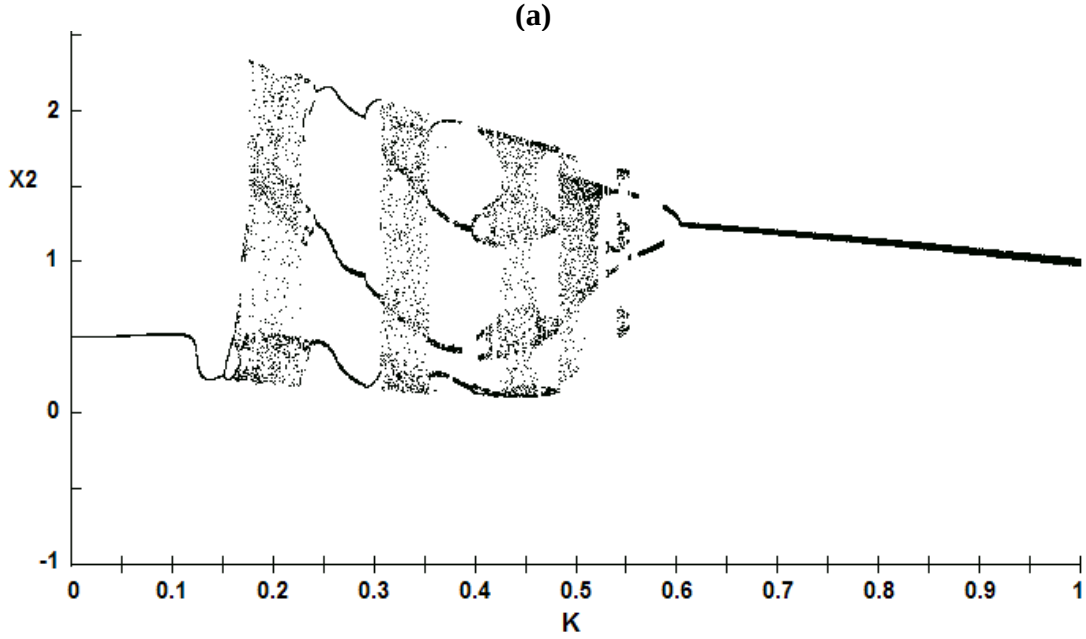
The bifurcation diagram in Figure 7.11 is drawn for the maximum amplitudes of X_2 and Y_2 in terms of the coupling constant K. The calculations have been done in the window of (K = 0, Y_2 = −5) to (K = 1, Y_2 = 4) for the initial conditions X_1 = 0.8, Y_1 = 0, and X_2 = 2, Y_2 = 0. Other parameters are set to fixed values of M = 0.8 and A= 0.5. Integration is over 5×10^4 points with step size 0.01, where the removed 500 transients has been repeated for each with K and ΔK = 0.0002. The maximum of Y_2 is determined based on the filtered values of Y_2. This is accomplished by following algorithm.

Algorithm

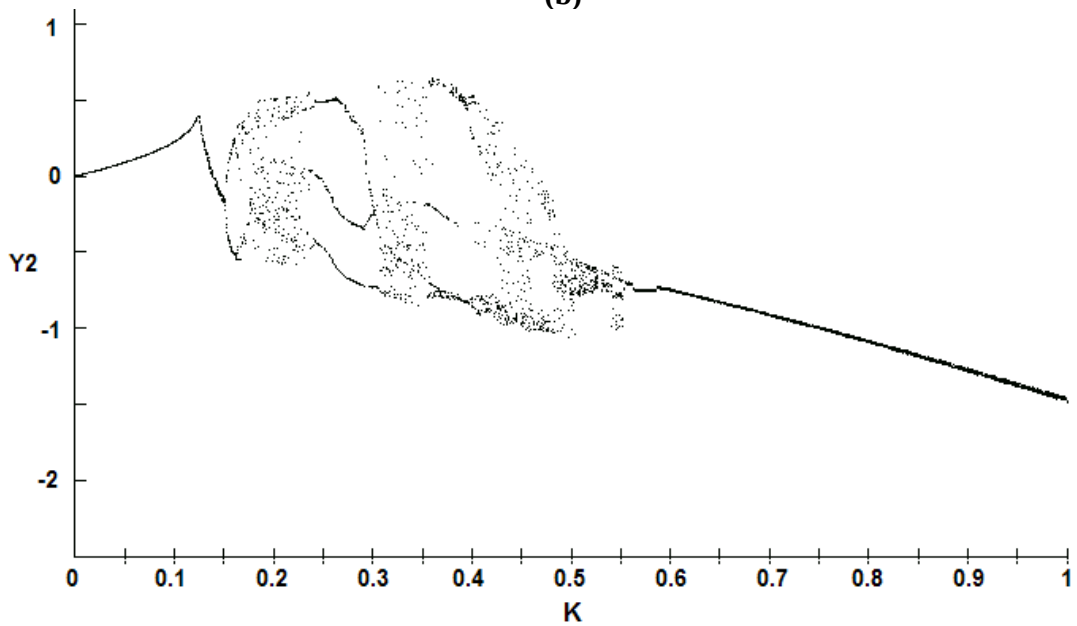
Set period (Tp) and control parameters and increment size, total iterations, transient responses, tolerance range
Remove transient responses
Initialize Time
Compute TTp = Time / Tp
Compute TTp – Integer (TTp)
If *above absolute difference is less than or equal to the Tolerance, then*
Set the partition for dynamical variables and Compute the dynamical states
Else
Select the next dynamical state
If *the dynamical state satisfies the partition criteria, then*
Select the dynamical state with corresponding control parameter
Else
Discard the dynamic state
End if

To initialize this algorithm, first the parameters are set to following numerical values. period (Tp) is 10, A = 0.5, M = 0.8, total number of iterations 50000, transient removal 500, K: [0 $\leq K \leq$ 1], ΔK=0.0002, Tolerance = 0.001, phase space partition: If $Y_1 \geq 0$ and $X_1 \geq 0$ and $X_1 \leq 0.1$, then the solution (K, X_2, Y_2) is selected. This algorithm samples the angular velocity of the Van der Pol oscillator at certain periodic intervals. The coupled system develops some periodic and chaotic regions if the coupling constant K is varied. Figure 7.11 (a) shows the result when the period is set to 2π. In order to increase the resolution and obtain a better bifurcation diagram, the period is adjusted to 10. Figures 7.11 (b) and (c) present a form of dynamic synchronization between X_2 and Y_2 showing the same dynamic states of periodic and chaotic motion for the same coupling constant K.

(a)



(b)



(c)

Figure 7.11 Bifurcation diagram for maximum amplitude X_2 vs. the coupling constant K [ΔK=0.0002] (a) T_p=2π (b) T_p=10 (c) Bifurcation diagram for maximum velocity amplitude Y_2. Same dynamical states occur for the same values of the coupling constant K. M = 0.8, A = 0.5, integration time-step 0.1 over 5×10^4 iterations after 500 transients, tolerance ~ 0.001.

In the same range of the bifurcation diagram we calculate the Lyapunov exponents spectrum (Figure 7.12) and LZ complexity (Figure 7.13) along the coupling constant *K*. The values of the first Lyapunov exponent coincide with the results of the LZ complexity and both agree with the bifurcation diagram (see Figures 7.13 and 7.14). The system shows very rich dynamic phases for various values of the parameters *M* and *A*. There are regions of periodic states and a periodic doubling state in wide chaotic states. The values of the Lyapunov exponents at the chaotic regions are positive. The values at the periodic regions are nearly zero. Thus, the periodic states are stable.

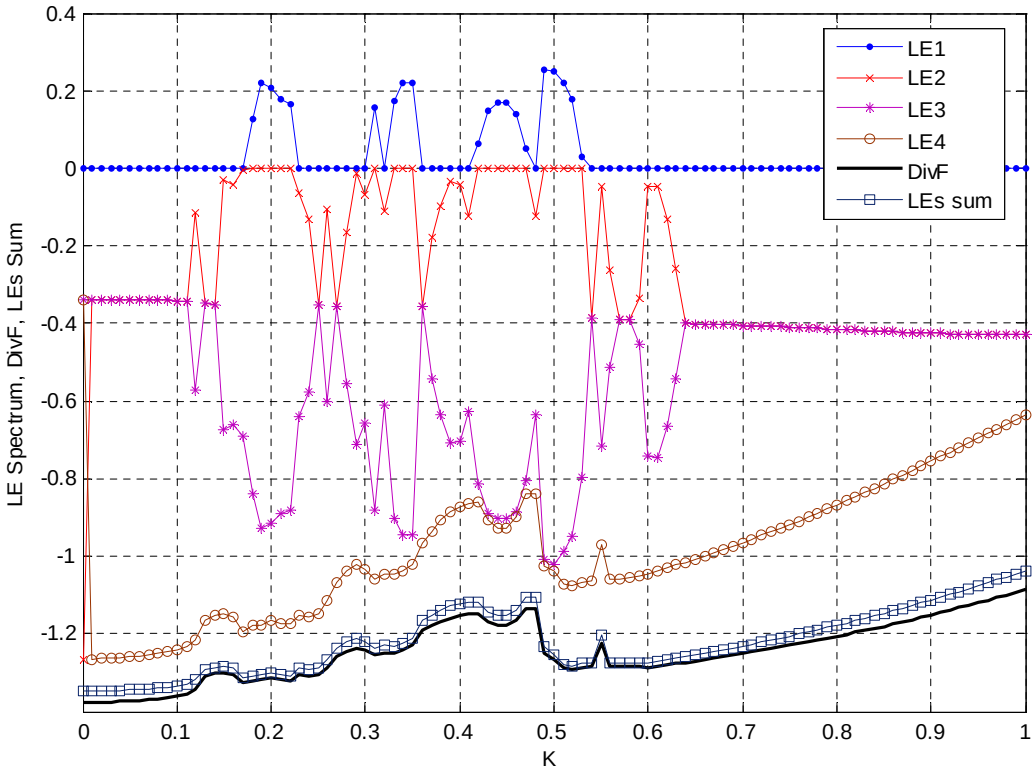


Figure 7.12 Lyapunov exponent spectrum vs. coupling constant *K* [ΔK=0.01]. *M*=0.8, *A*=0.5, integration time-step 0.1, over 2×10^5 iterations after 10^4 transients, GSR step 1.

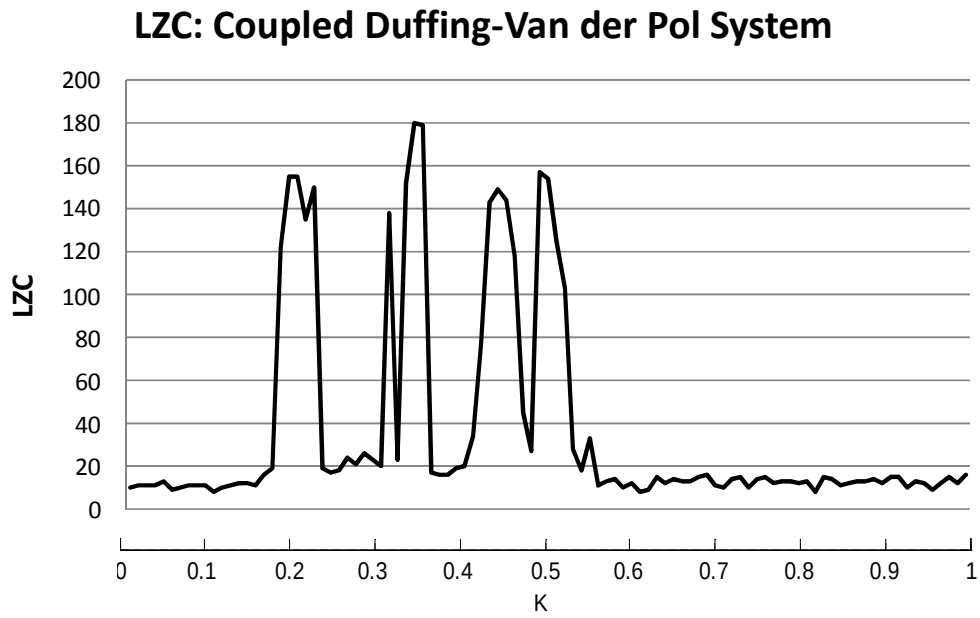


Figure 7.13 Lempel-Ziv complexity in terms of coupling constant K [ΔK=0.01]. M=0.8, A=0.5, integration time-step, 0.1, buffer size 32700, after 5×10^3 transients.

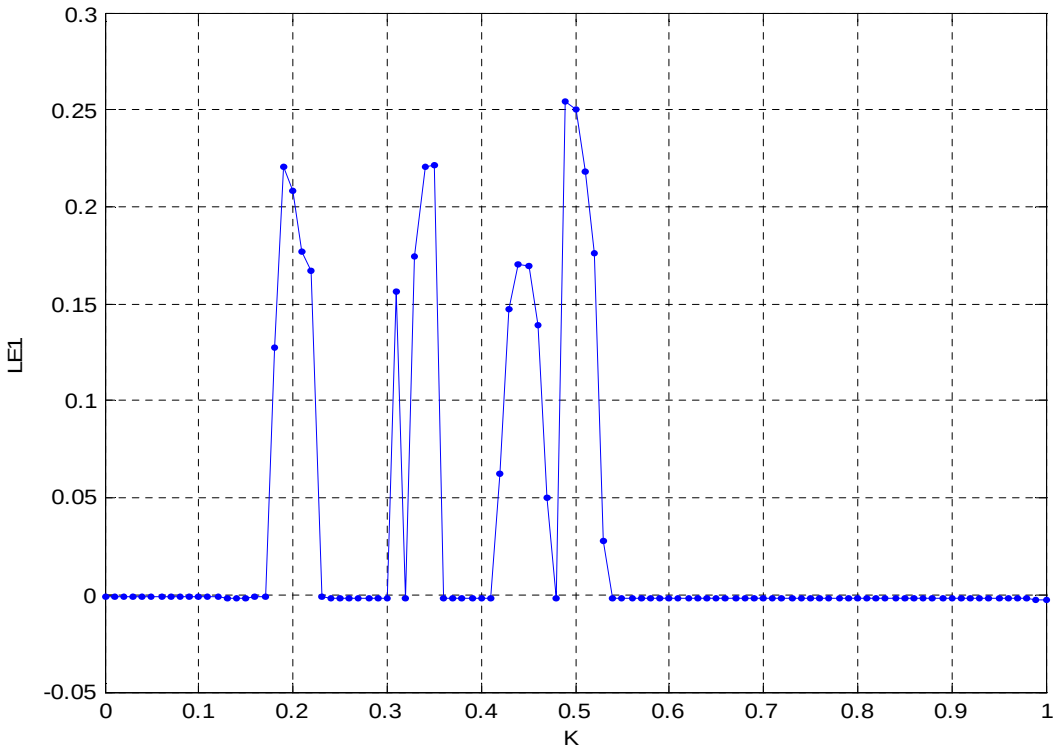


Figure 7.14 First Lyapunov exponent in terms of coupling constant K [ΔK=0.01], M=0.8, A=0.5, integration time-step, 0.1, over 2×10^5 iterations after 10^4 transients, GSR step 1.

To formalize the steps required to compute system characteristic information across all possible values of the control parameters and get more understanding of the associated effects, we calculate another combined phase diagram (bifurcation diagram confirmed by Lyapunov exponent spectrum). The coupling constants are K = 0.5 and M = 0.4. The dynamics is subject to one of the attractors based on their relative strengths. In this case, the system shows again different states from a period-one state due to the limit-cycle attractor and damping states due to the point attractor. Chaotic states and multi-period states are also found (Figures 7.15, 7.16 and 7.17). It is found that the states get closer to stable states as the parameter A increases from 0.17 to 1.

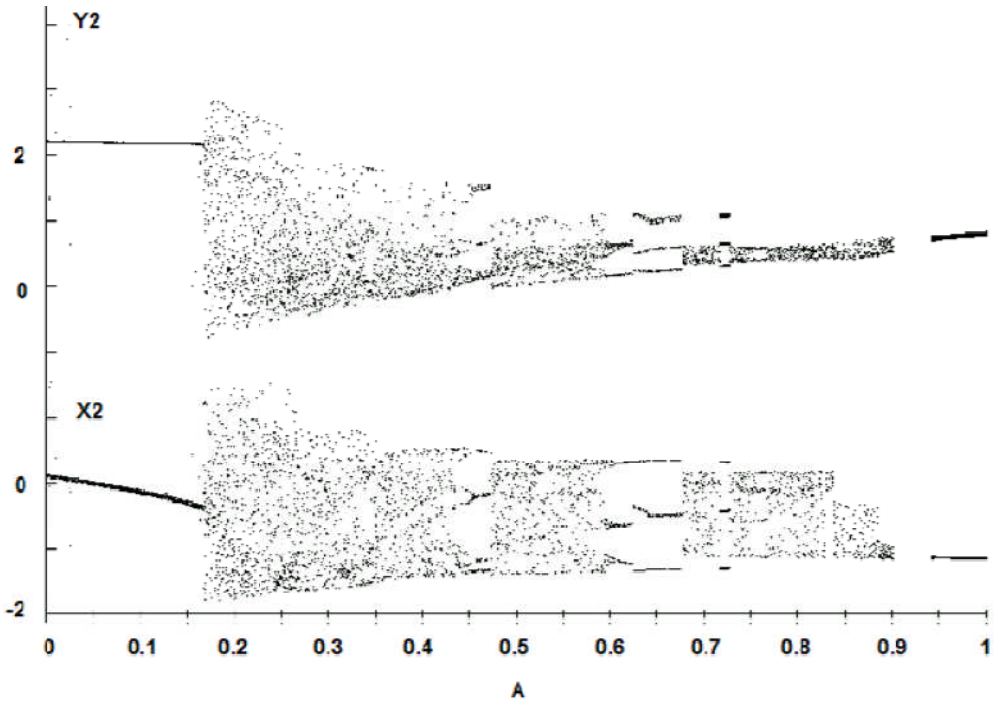


Figure 7.15 Bifurcation diagram for maximum amplitude X_2 and maximum velocity amplitude Y_2 vs. the coupling constant A [ΔA=0.01], T_p=10, M=0.4, K=0.5, integration time-step, 0.1, over 5×10^4 iterations after 500 transients with tolerance ~ 0.001.

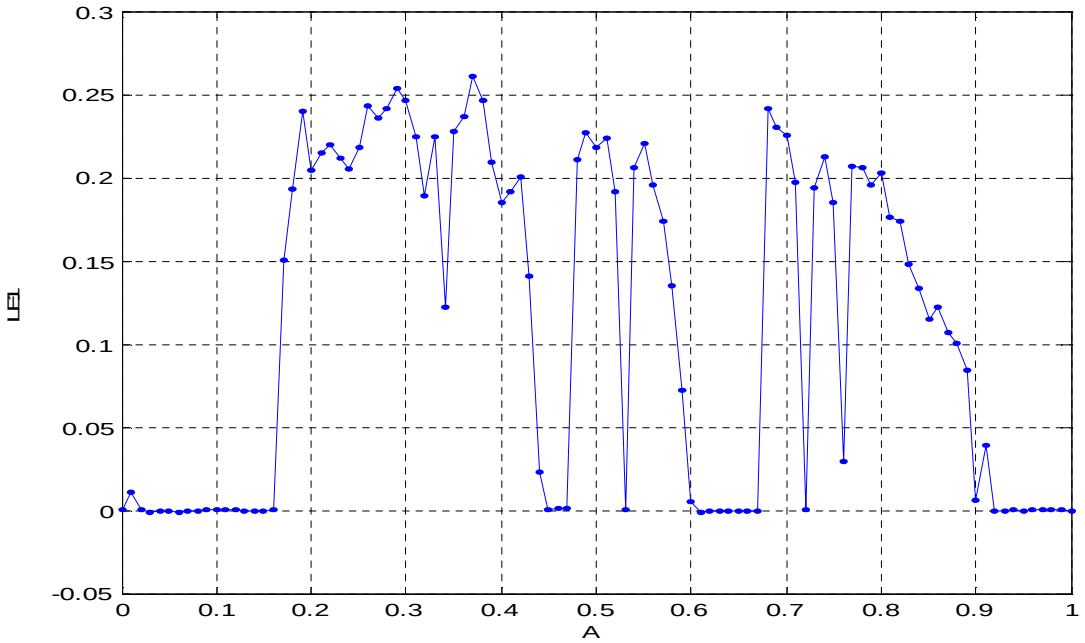


Figure 7.16 First Lyapunov exponent vs. coupling constant A [ΔA=0.01], M=0.4, K=0.5, integration time-step, 0.1 over 3×10^5 iterations after 10^4 transients, GSR step 1.

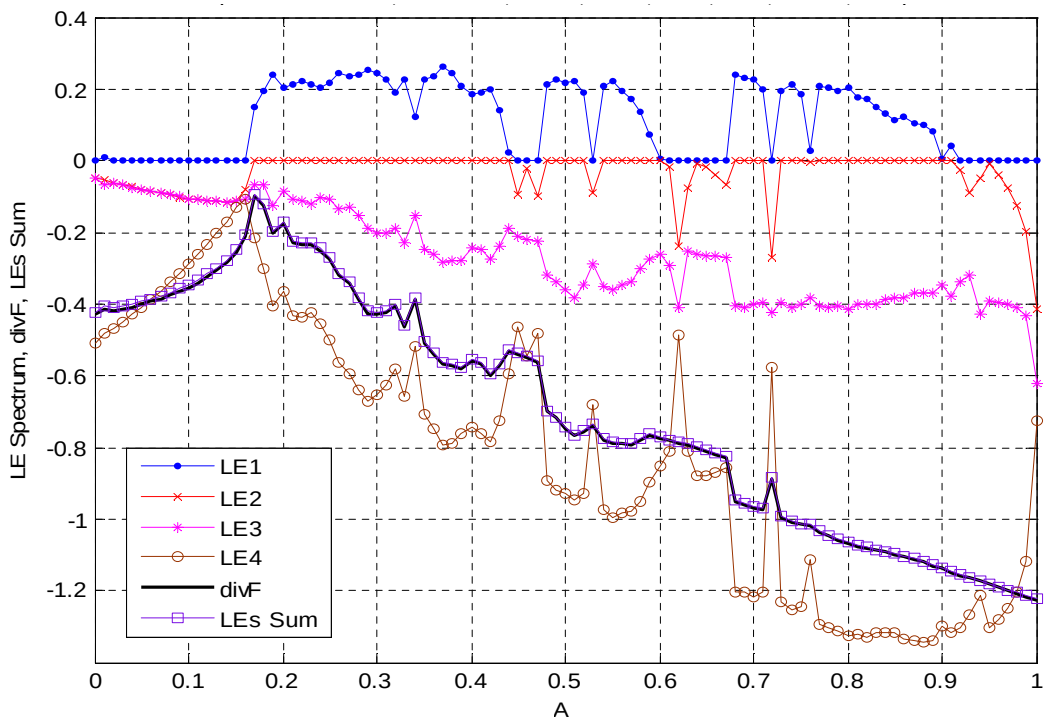


Figure 7.17 Lyapunov exponents spectrum in terms of coupling constant A [ΔA=0.01], M=0.4, K=0.5, integration time-step 0.1, 3×10^5 iterations, 10^4 transients, GSR step 1.

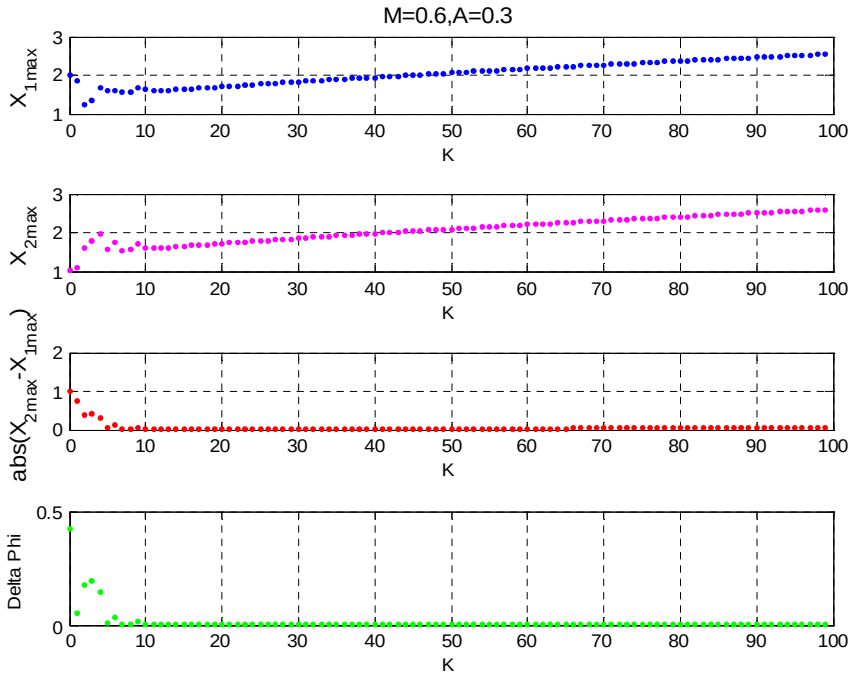
7.3 Coupled Non-Homogeneous Chaos Synchronization

One of the interesting phenomena in coupled nonlinear oscillators is synchronization which has fundamental importance in secure communication systems, laser dynamics, electronic circuits and biological systems. In this section, we explore the chaos synchronization in a coupled non-homogeneous oscillating system. This is characterized by the loss of exponential instability in the traverse direction through system interaction. Additionally, the dependence of the synchronization properties of the system with respect to the intensity of the coupling constant is examined.

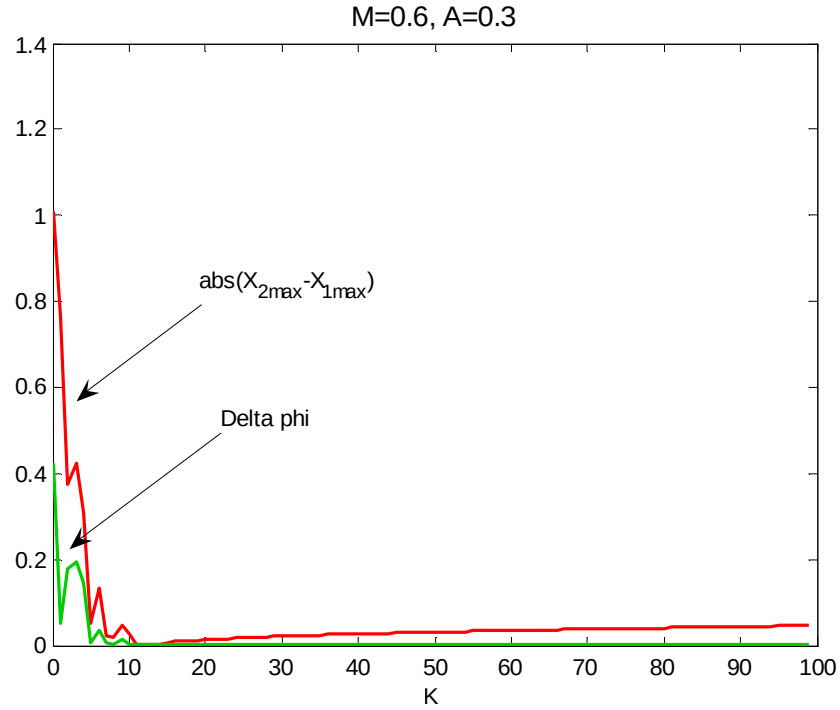
Several types of synchronizations representing different degrees of correlation between the interacting systems have been identified. These are referred to as complete synchronization, generalized synchronization, lag synchronization, frequency synchronization and phase synchronization. Lag synchronization appears as a coincidence of the states of two systems, $Y_1(t) = Y_2(t+\tau)$, when shifted in time. A weaker form of synchrony between oscillators is phase synchronization (PS). The coupled system can undergo oscillations of widely different magnitudes, even chaotic ones, but with equal phases, $\varphi_1(t) = \varphi_2(t)$. In other words, phases of two chaotic oscillators lock to each other while their amplitudes remain uncorrelated and sustain an irregular motion. For nearly identical systems, that is, when they present a small parameter mismatch, this can be true up to a possibly small constant: $\varphi_1(t) \approx \varphi_2(t)$. Two systems exhibiting phase synchronization may not present complete synchronization. It may also happen that, even when the phases themselves are not equal, their time rates might be equal, which characterizes frequency synchronization $\omega_1 \approx \omega_2$ [109]. If the oscillator amplitudes are behaving periodically instead of chaotically, phase synchronization is nothing but mode-locking, or a commensurability between frequencies $m\Omega_1 - n\Omega_2 = 0$, where m and n are positive integers.

If the coupled systems are identical, a synchronization manifold Γ is defined through the conditions $X_1(t) = X_2(t)$, $Y_1(t) = Y_2(t)$. The complete synchronization state exists when it is asymptotically stable for all possible trajectories of the driving oscillator, either periodic or chaotic, against small displacements of the synchronized trajectories. Complete synchronization of oscillators implies that the time series of the corresponding dynamical variables of the subsystems coincide completely. This happens if the systems oscillate in phase with same amplitude. Generalized synchronization is introduced for drive-response systems. It is characterized by the presence of a functional relationship between the amplitudes of the two states of the coupled oscillators' responses, and that may occur even for non-identical systems [110, 111, 112].

Using the aforementioned analytical definitions, various synchronization phenomena are studied in an autonomous four-dimensional vector field model represented by coupled oscillators (Eqs. 7.3). Results show that phase synchronization occurs when the coupling constant K is less than 100. Increasing the coupling constant causes the phase synchronization to be replaced by complete synchronization as well as lag synchronization. Complete synchronization happens when the coupling constant is very large. To determine the range of K, we change the coupling constant for different values of the parameters M and A. One of the results is shown in Figure 7.18 where the maximum oscillators amplitudes, X_1 and X_2, the absolute difference $|X_{1\max} - X_{2\max}|$ and the phase difference $|\Delta\varphi|$ are plotted when $A = 0.3$, $M = 0.6$ over 10^3 solutions with time-step 0.01, after 2×10^3 transients. State phase is defined as $\varphi_i = \tan^{-1}(Y_{i\max} / X_{i\max})$. The system approaches complete synchronization asymptotically. Note that the phase synchronization is sustained at strong coupling constant from 10 to 100.



(a)



(b)

Figure 7.18 (a) Plots of maximum Van der Pol (X_1) and Duffing (X_2) oscillators amplitude (b) absolute maximum amplitude difference $|X_{1\max} - X_{2\max}|$ and phase difference $|\Delta\varphi|$ vs. K, [ΔK = 1], A = 0.3, M = 0.6, 10^3 solutions with time-step 0.01, 2×10^3 transients.

In the large coupling region, the synchronization states of the system are determined by the relative strengths of the system parameters A and M. In the series of trials demonstrated in Figure 7.19 through 7.29, the phases of complete and phase synchronization are recognized. The results are listed in Table 7.1. Note that the occurrence of complete synchronization small values of coupling constant, K=5, for M = 1 and A = 3. In addition, for $A \geq M$, phase synchronization occurs at relatively small values for coupling constant. In other word, the separation of two synchronization phases is dependent on only the values of the ratio of A to M if M is not too large, that is, $M < 6$. Our results confirm that the complete synchronization and the phase synchronization states are well separated by the condition at which the value of A is almost equal to M. Observe that, by increasing A and M, system exhibits complete synchronization at a larger value of K.

Table 7.1 Synchronization states parameter values

M	***A***	***K* for PS**	***K* for CS**
1	1	5	40
1	4	6	15
1	5	7	15
1	3	5	5
3	1	21	100
3	3	10	40
4	4	13	65
4	5	14	27
5	1	53	85
5	5	16	80
6	5	20	38

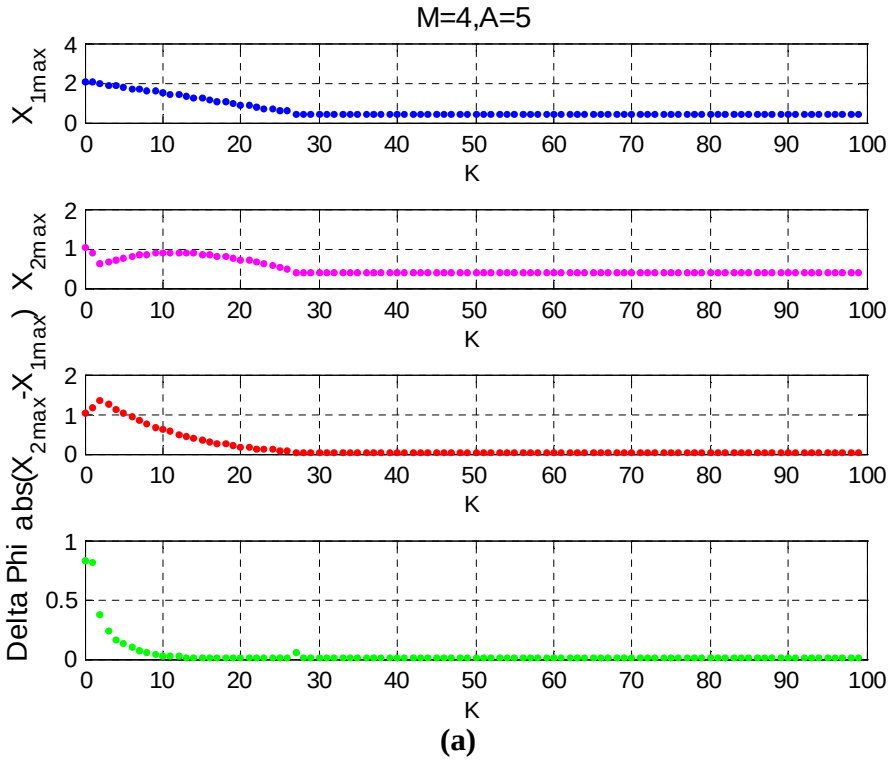
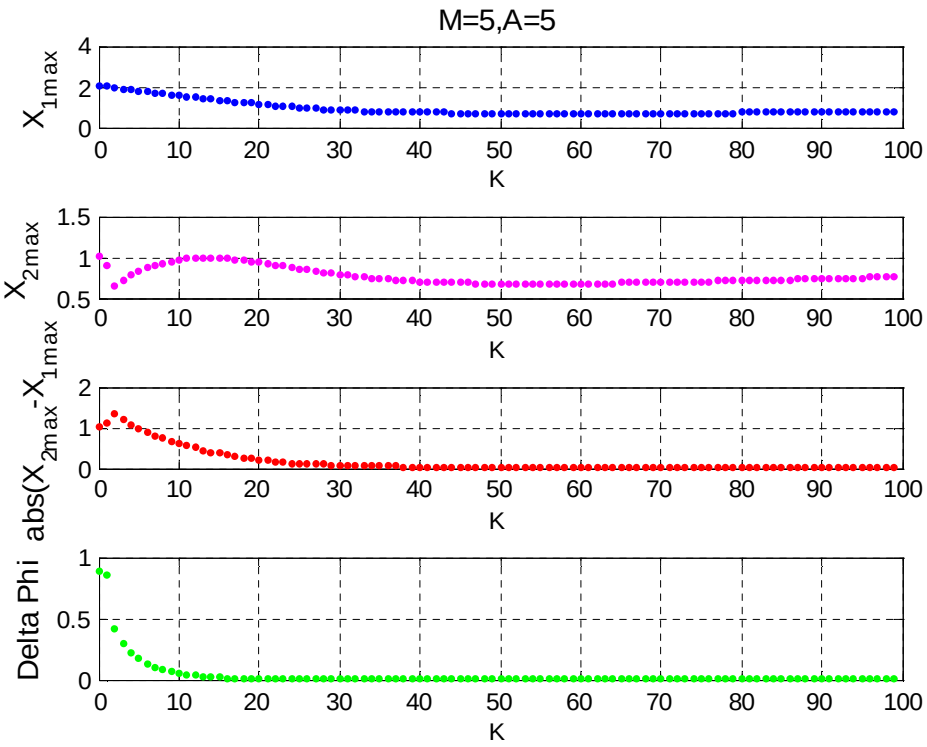
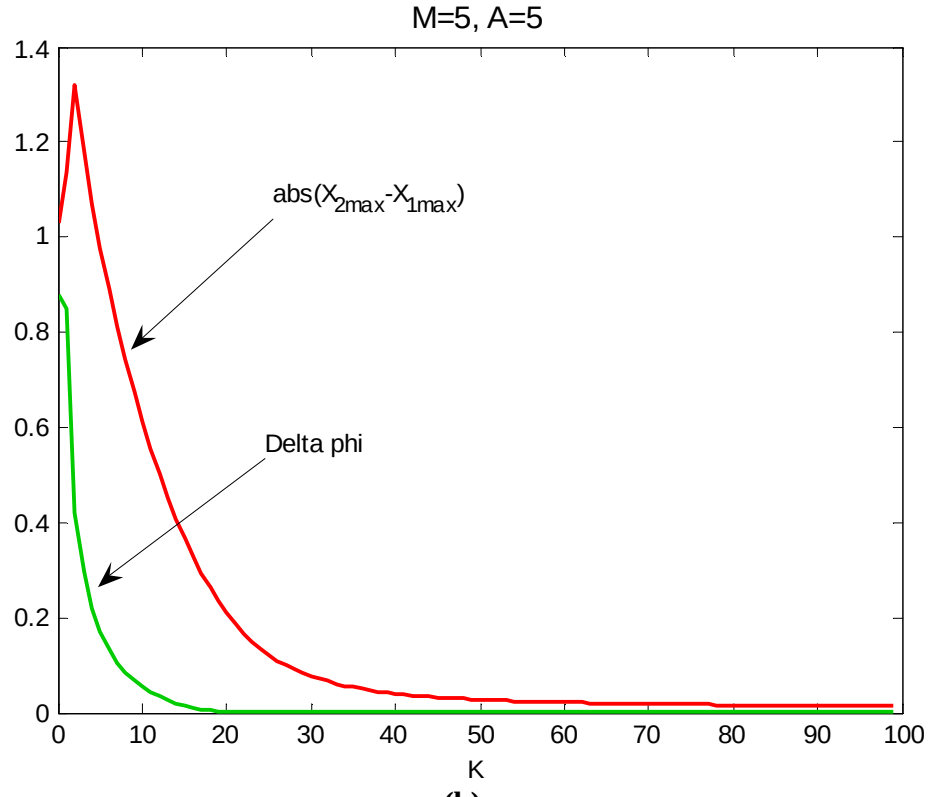


Figure 7.19 (a) $X_{1\text{Max}}$ and $X_{2\text{Max}}$ (b) $|X_{1\text{max}} - X_{2\text{max}}|$ and $|\Delta\varphi|$ vs. K for $M = 4$, $A = 5$.

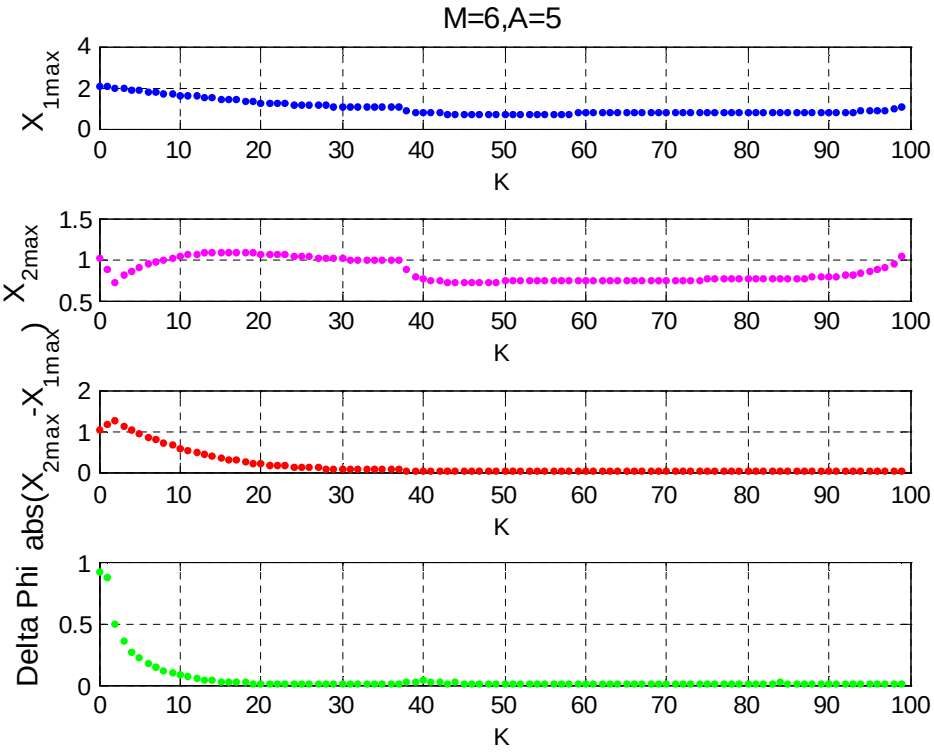


(a)

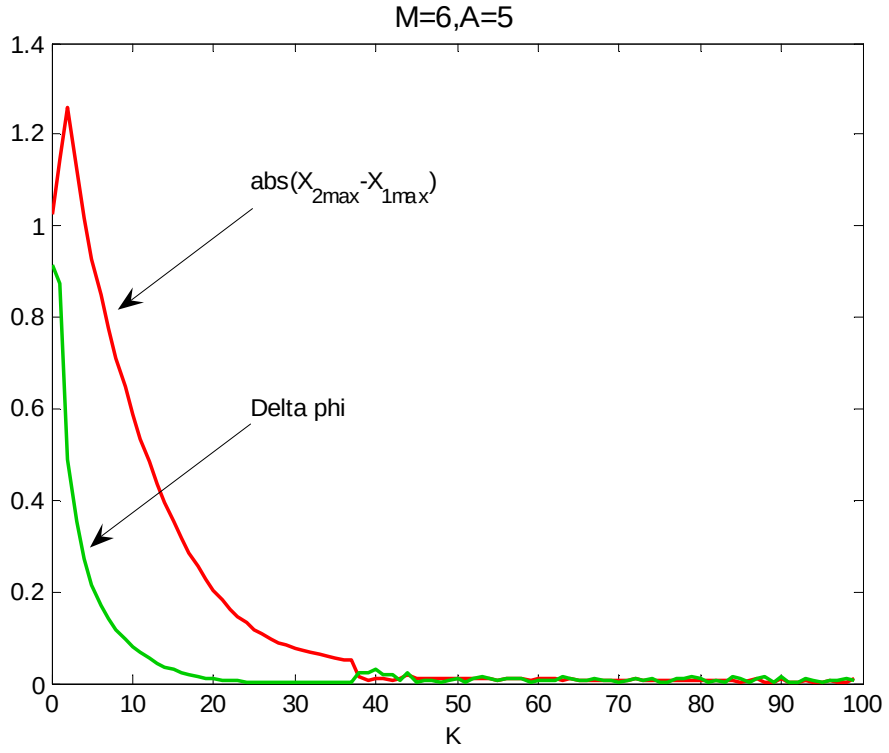


(b)

Figure 7.20 (a) $X_{1\max}$ and $X_{2\max}$ (b) $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 5, A = 5$.

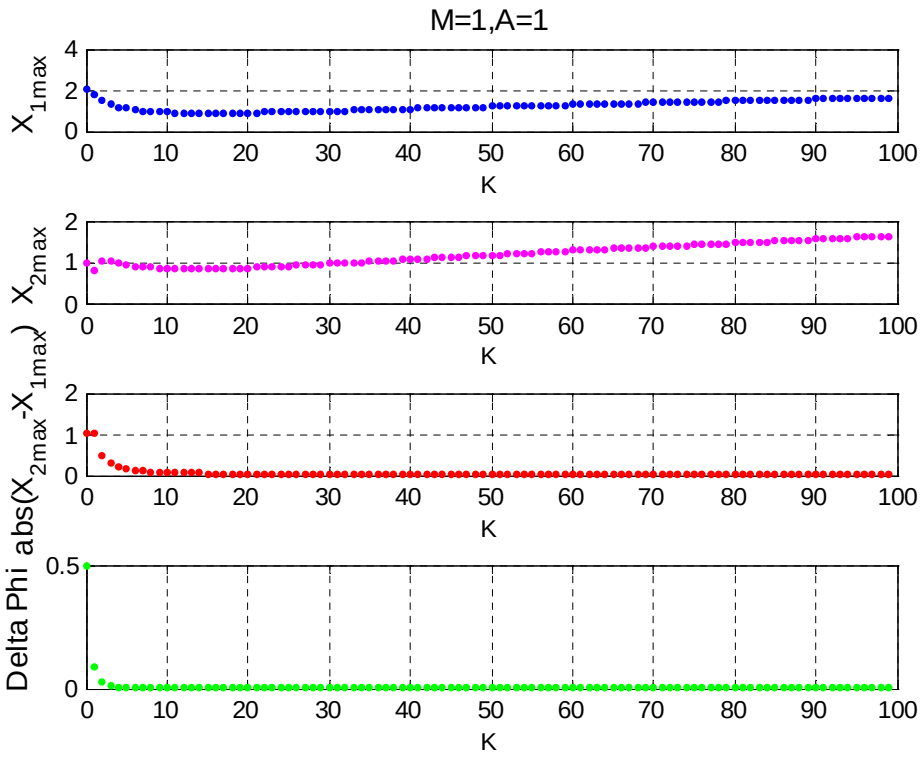


(a)



(b)

Figure 7.21 (a) $X_{1\max}$ and $X_{2\max}$ (b) $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 6, A = 5$.



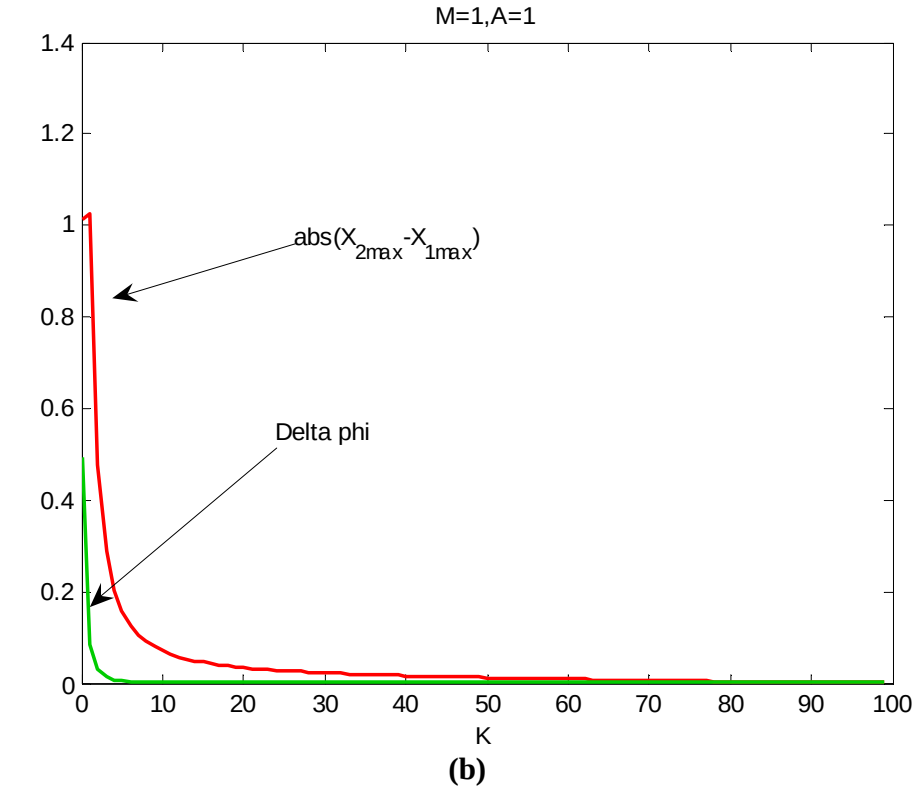
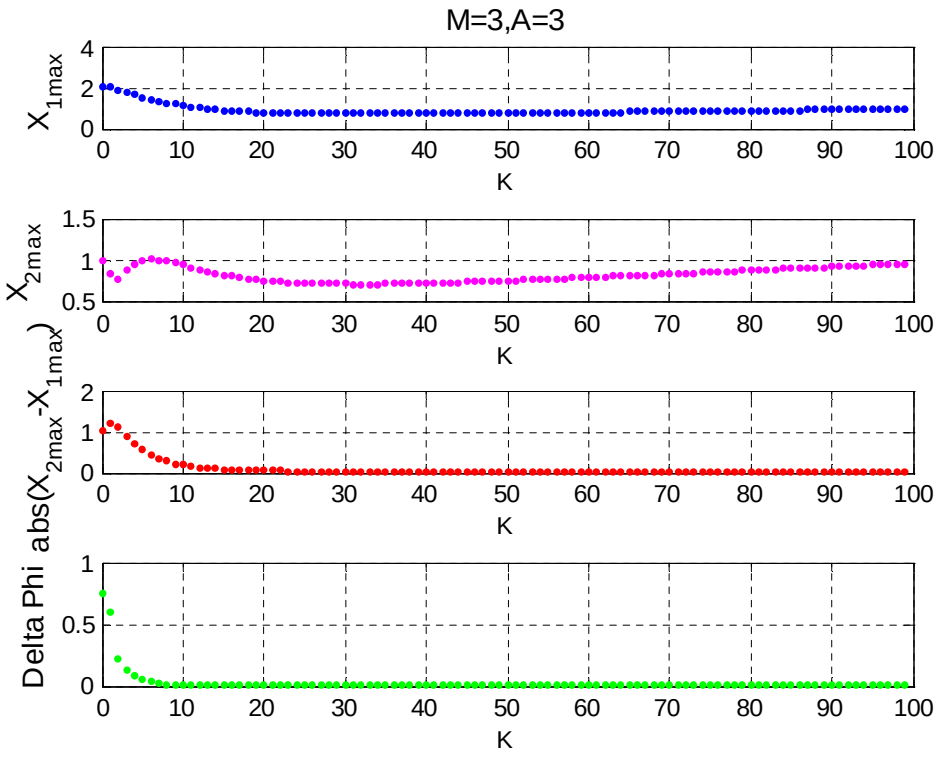
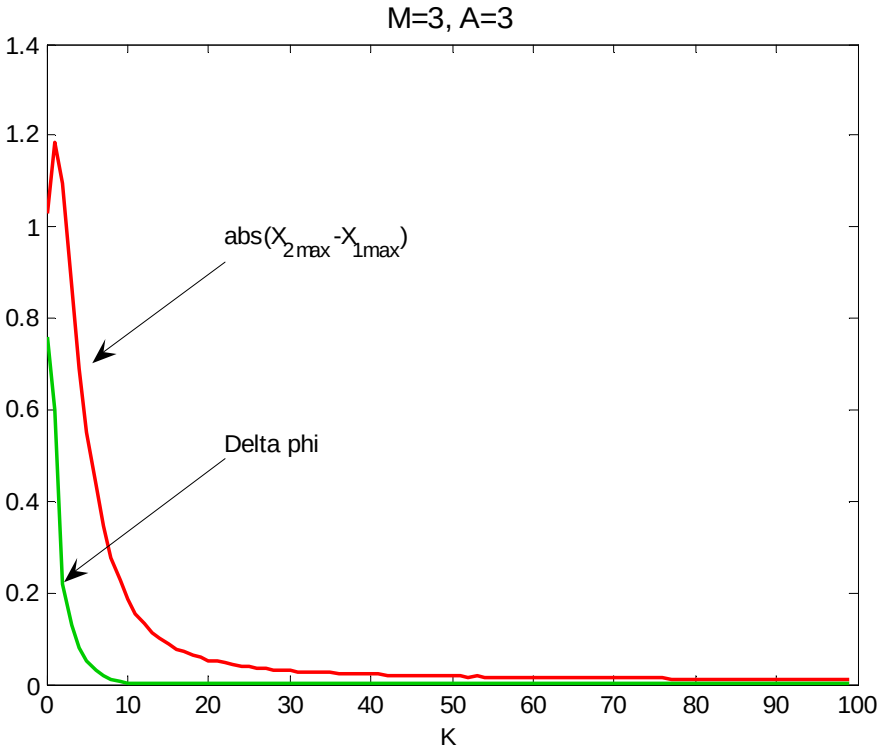


Figure 7.22 (a) $X_{1\max}$ and $X_{2\max}$ (b) $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 1, A = 1$.

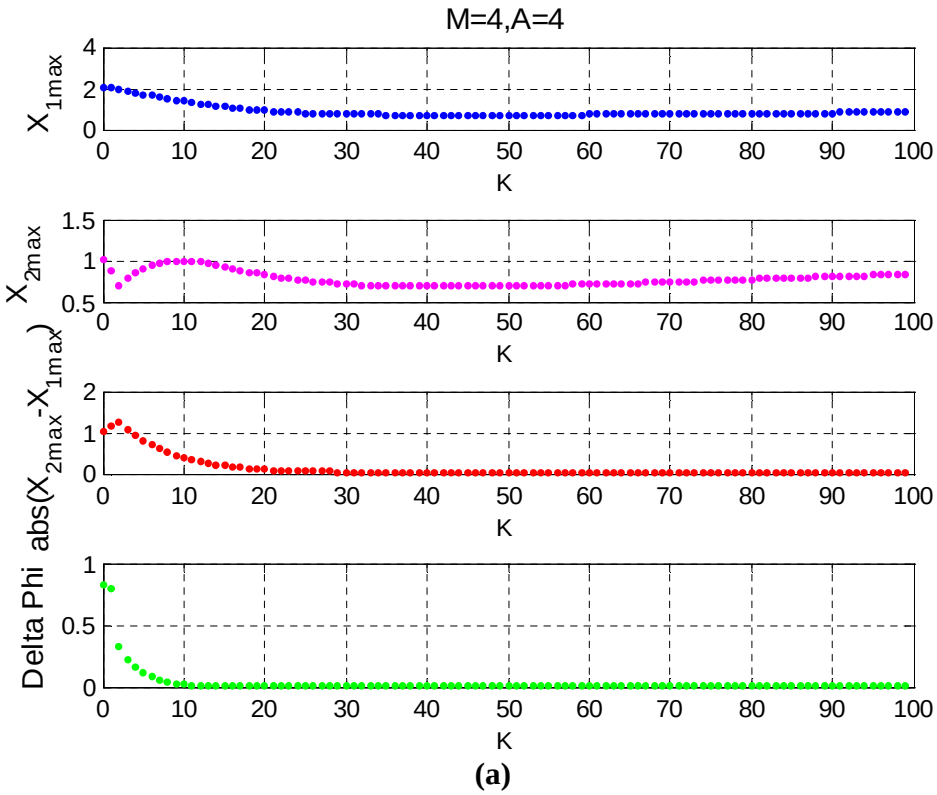


(a)

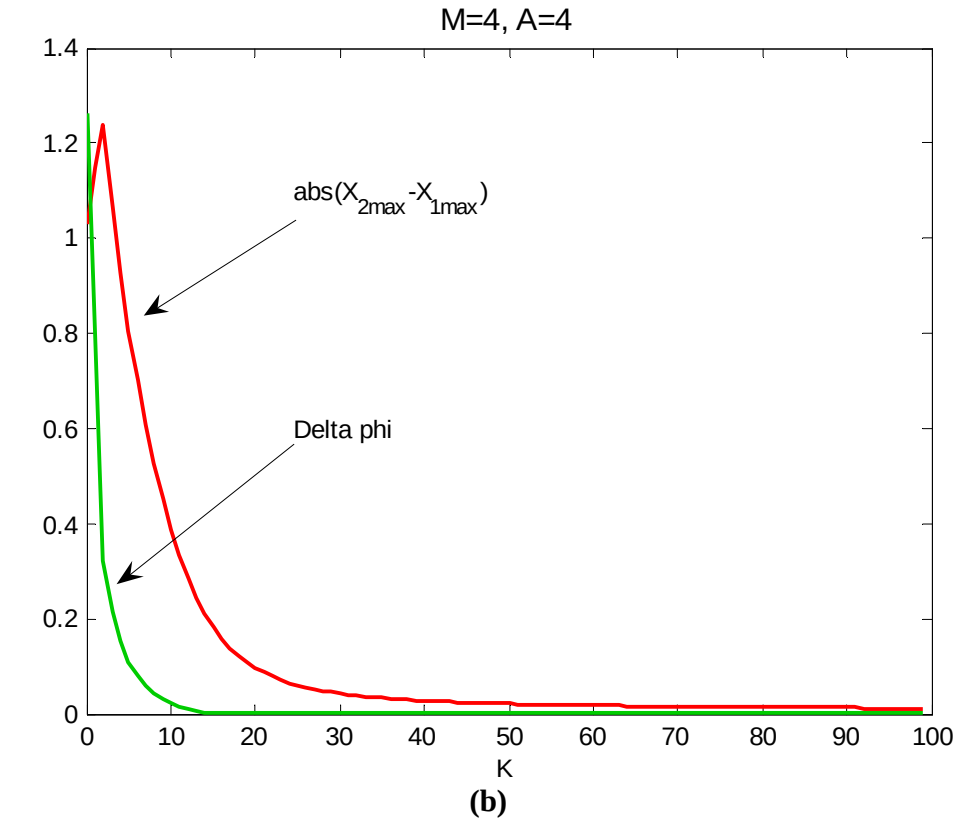


(b)

Figure 7.23 (a) $X_{1\max}$ and $X_{2\max}$ (b) $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 3, A = 3$.

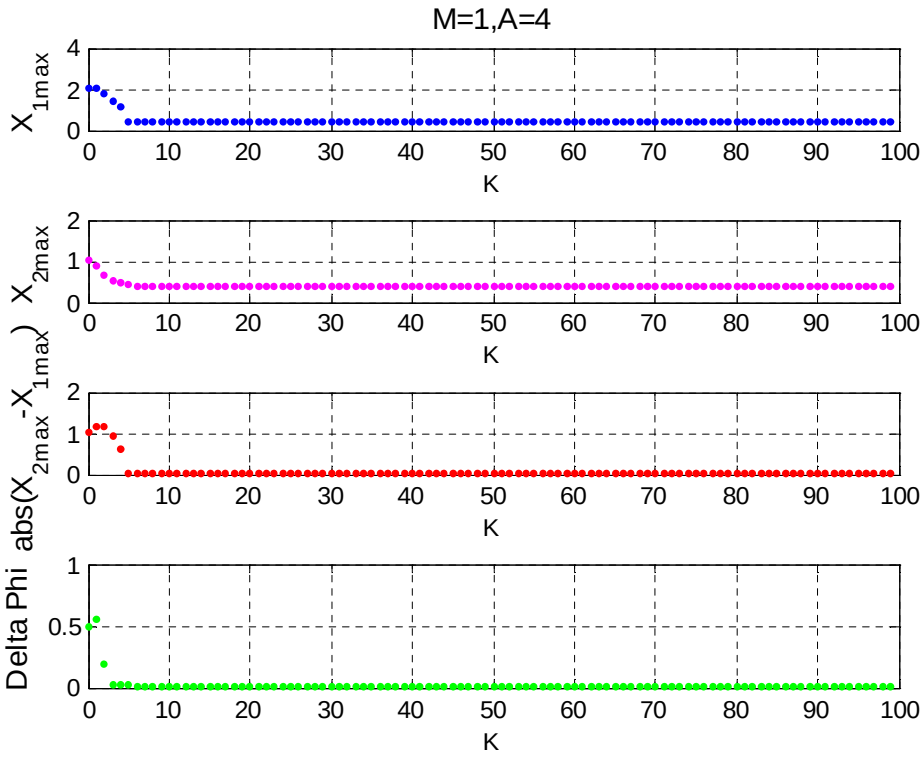


(a)

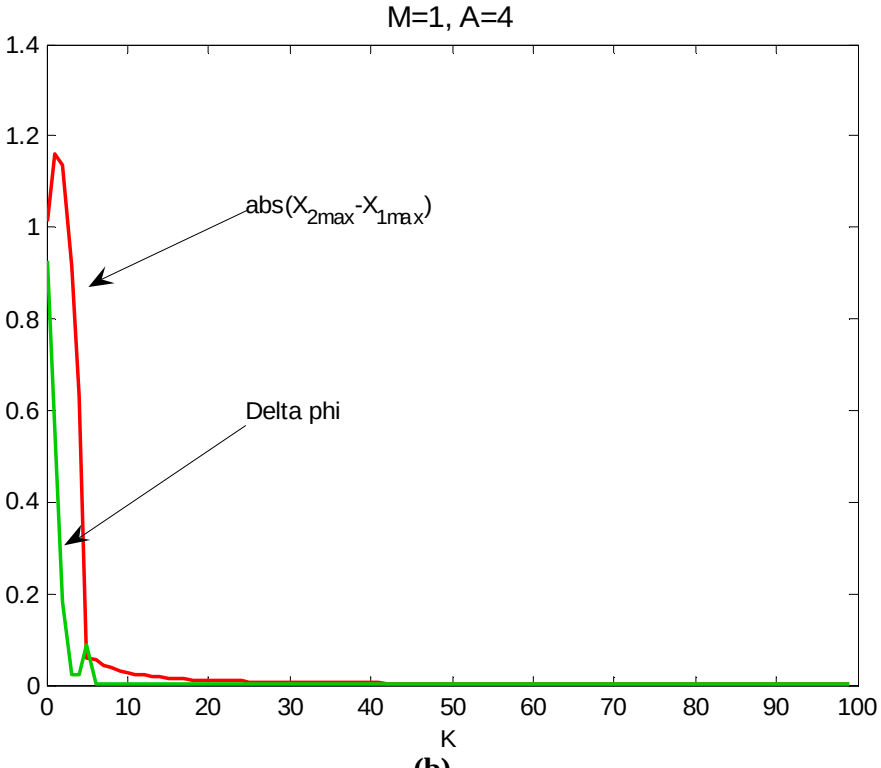


(b)

Figure 7.24 (a) $X_{1\max}$ and $X_{2\max}$ (b) $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 4, A = 4$.



(a)



(b)

Figure 7.25 (a) $X_{1\max}$ and $X_{2\max}$ (b) $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 1, A = 4$.

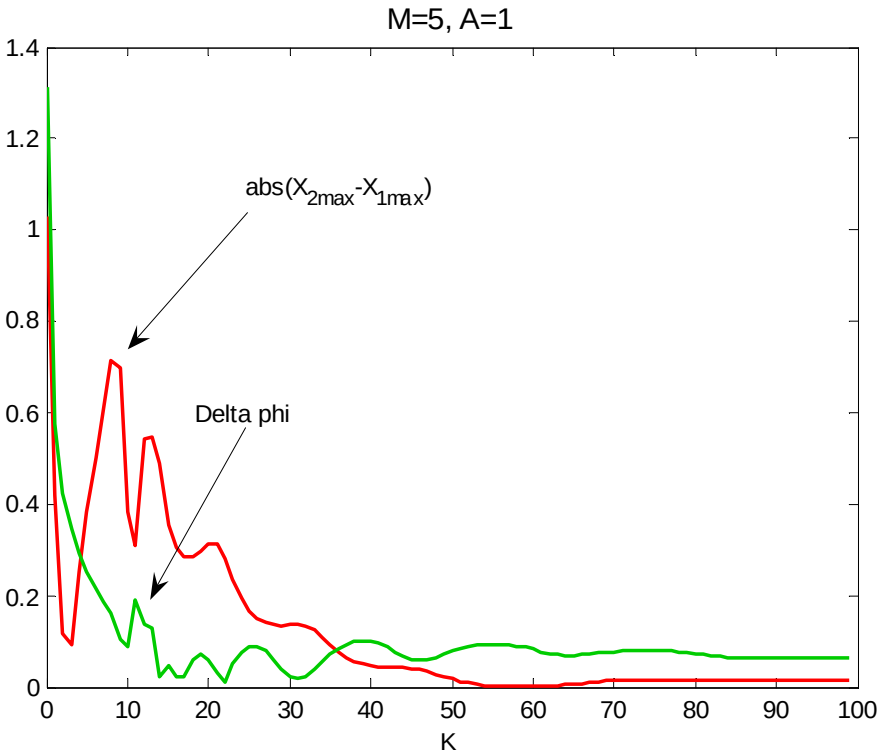


Figure 7.26 $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 5$, $A = 1$.

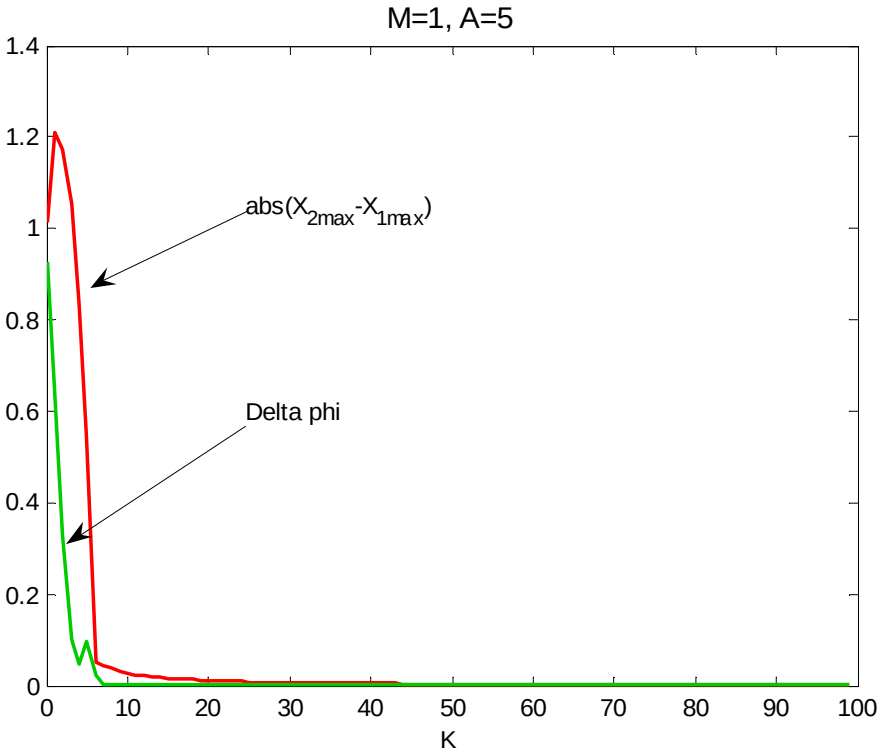


Figure 7.27 $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 1$, $A = 5$.

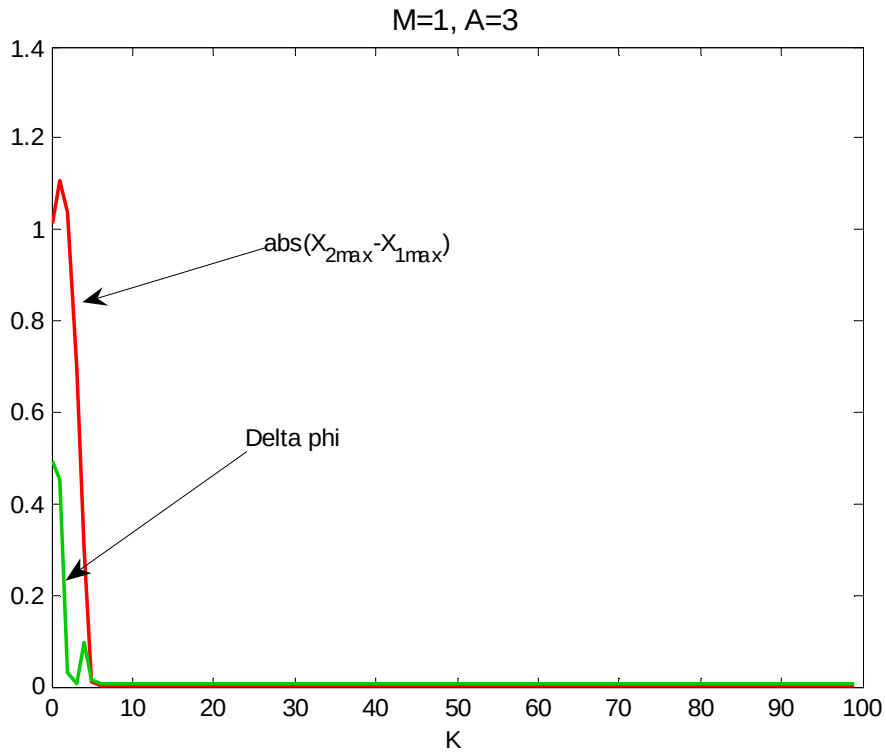


Figure 7.28 $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 1, A = 3$.

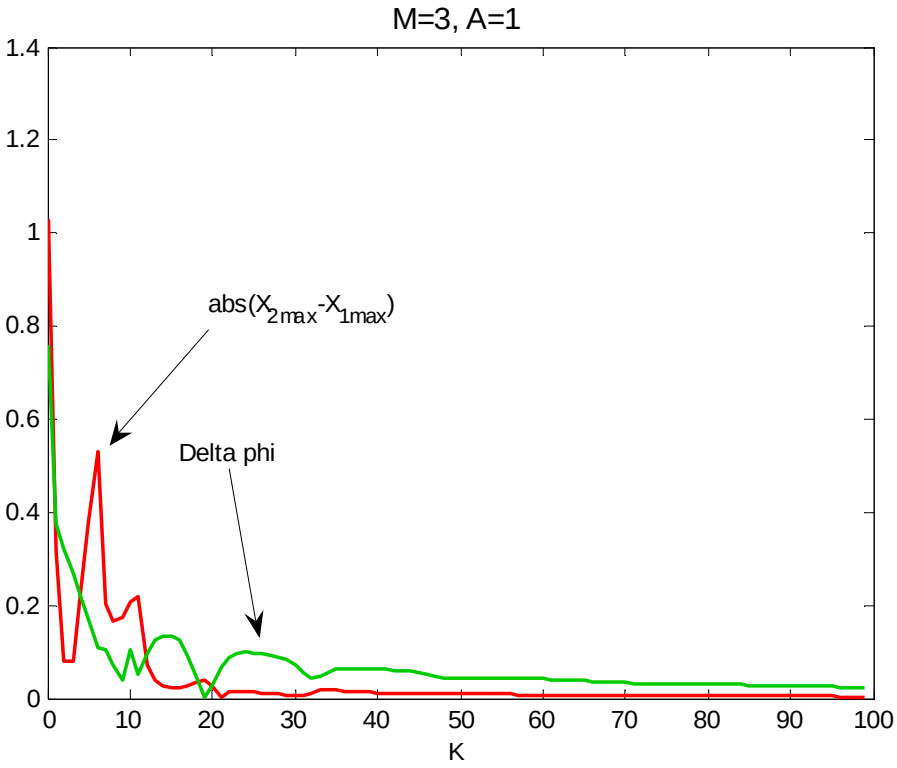


Figure 7.29 $|X_{1\max} - X_{2\max}|$ and $|\Delta\varphi|$ vs. K for $M = 3, A = 1$.

We calculate another phase diagram in parameter space for a stronger coupling constant. The simulated system has K = 100 and the phase computation uniformly distributed on the field that spans 100 points by 100 points on the M and A parameter values grid. A summary of the system details is provided in Figure 7.30. The phase diagram shows only two different states. The phase synchronized states are on flat part of the diagram. The characteristics of various chaotic and periodic states are observed in the control parameter space from non-synchronization to phase synchronization in this coupled oscillator. There exists a special locking regime in which a state shows maximal periodicity. This is quasi-periodic phase synchronization (QPPS) state. We did not find any relation between positiveness in the maximum Lyapunov exponent and occurrence of QPPS synchronization states.

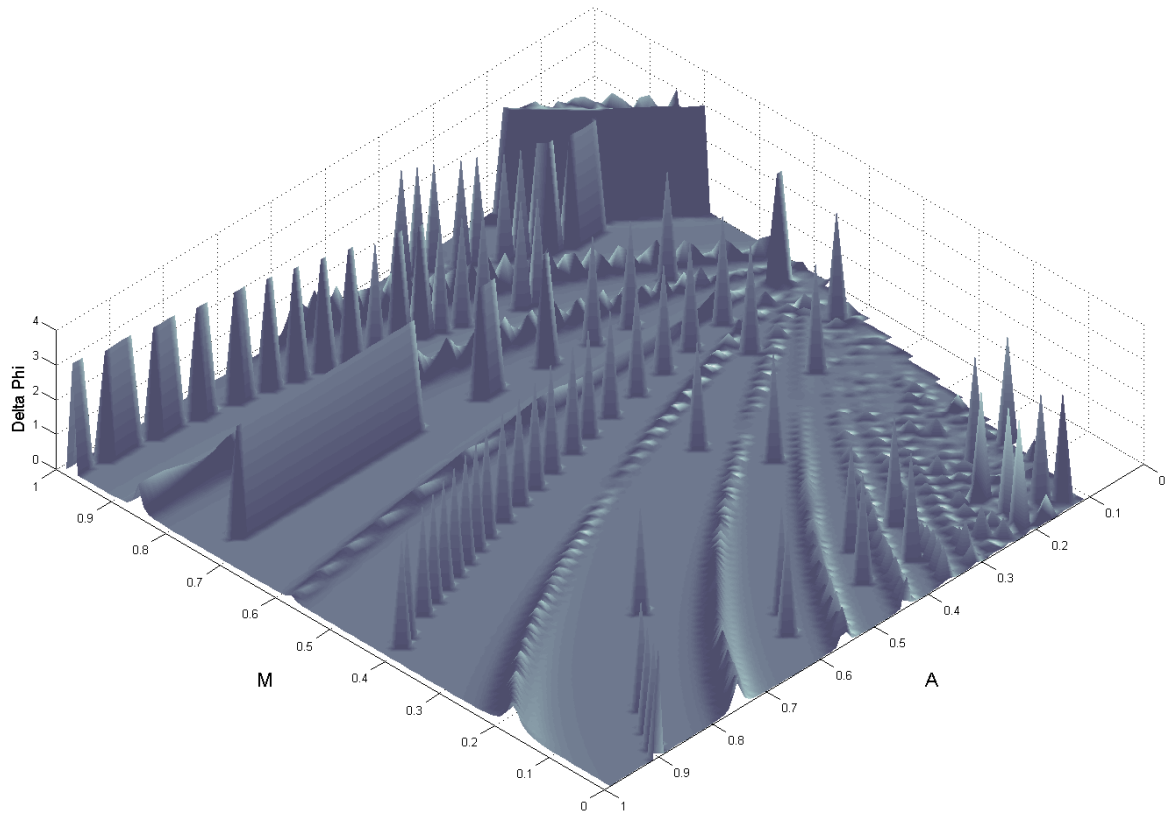


Figure 7.30 $\Delta\varphi$ plot for phase synchronization when coupling constant is K = 100.

Figures 7.31 and 7.32 show two different synchronizations at large coupling constant (K = 100) near $A = M$ in terms of time series. Solid blue line represents the VP oscillator while red dashed-line is DF oscillator. It demonstrates the complete and lag synchronization states as well as the damping state for various relative strengths of A and M near the ratio $A/M \sim 1$.

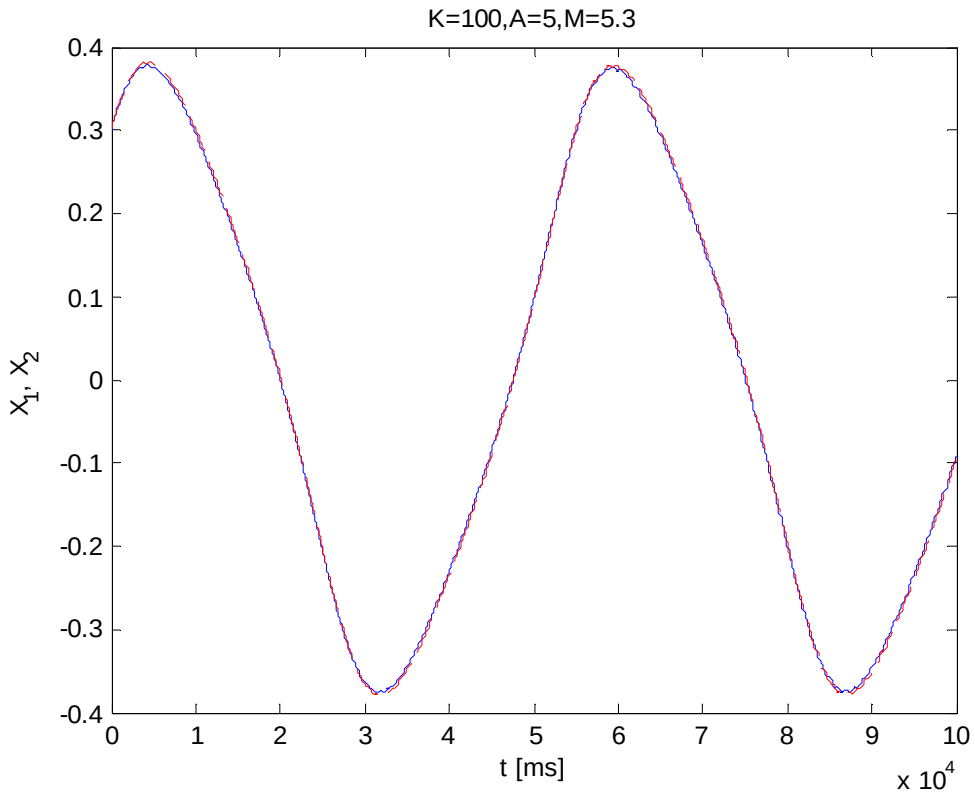


Figure 7.31 Complete synchronization state at A=5, M=5.3, K = 100.

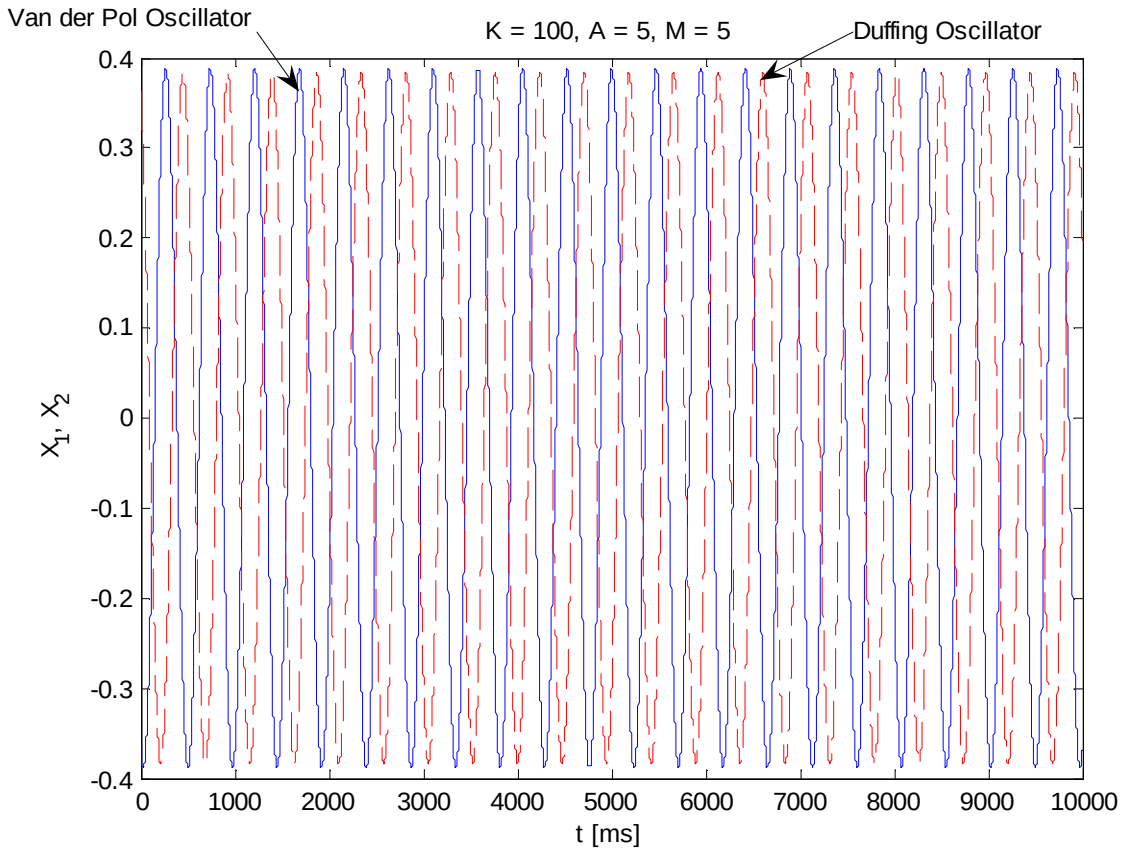


Figure 7.32 Lag synchronization state at A=5, M=5 when coupling constant is K = 100.

7.4 Conclusion

A system model for a non-homogeneous system of coupled nonlinear oscillators was developed and several simulations are performed. The system dynamics is studied by varying the system control parameters. Two dynamic phase diagrams are constructed in terms of the control parameters. This is done by computing the Lyapunov exponent spectrum and selecting the positive values as the indicator of chaos. In determining periodic regions the bifurcation diagrams are used. As the coupling constant gets larger, the dynamics of the system becomes simpler. The simulation results provide evidence for three types of synchronization and confirm that our coupled non-homogeneous chaotic oscillators exhibit three forms of synchronizations:

phase, lag and complete. The transitions between various forms of synchronizations occur for different coupling parameter strengths. Complete synchronization is unlikely to be observed in practice, due to the non-homogeneity of the coupled oscillators, but our simulation predicts theoretically its occurrence.

CHAPTER 8

FIBONACCI QUASI-PERIODIC SYSTEM LEMPEL-ZIV COMPLEXITY

8.1 Fibonacci Quasi-Periodic System

In previous chapters the results from algorithmic complexity analysis are compared with the results from a Lyapunov metrics computation for various nonlinear systems. We have observed that using these two metrics, we can distinguish order from disorder. In the present chapter we use the same metrics to study the complexity measure of a Fibonacci map as a quasi-organized system. We show analytically and numerically that for a system like a Fibonacci map, complexity grows with the length of the data block. The order of organization for this quasi-organized system is increasing logarithmically with the evolutionary length of the system. We conclude that using a normalized algorithmic complexity measure, as a pattern classifier, gives a non-zero value less than one for chaotic responses. For periodic and quasi-periodic responses, as the data string grows, the complexity approaches zero, with a higher deceasing rate is observed for periodic responses.

A Fibonacci system as a quasi-periodic dynamical system defined by the following map:

$$F:\begin{cases} X_{n+1} = X_n + Y_n \\ Y_{n+1} = X_n \ . \end{cases}$$

To show analytically that the algorithmic complexity of a block of a binary Fibonacci sequence increases logarithmically with the length of the block, we first need the following definition.

Definition The subsequence $A_i A_{i+1} \ldots A_{i+m}$ with length $m \le l$, from the sequence $A_1 A_2 \ldots A_i \ldots A_{i+m} \ldots A_l$ is called a block.

For the Fibonacci map consider the string $\{f_k\}_{k=0}^{N}$, $f_k = f_{k-1} + f_{k-2}$ where $k \ge 2$. Fibonacci blocks are constructed by concatenating two previous blocks $F_k = F_{k-1} \oplus F_{k-2}$ starting from unit blocks, i.e., block zero $F_0 = 0$ and block one $F_1 = 1$. We list the Fibonacci blocks in the following table. Figure 8.1 shows the structural relation between each block and its building blocks:

Table 8.1 List of Fibonacci blocks

Index	**Block length**	**Block**
n=0	$f_0=1$	$F_0=0$
n=1	$f_1=1$	$F_1=1$
n=2	$f_2= f_1+ f_0$	$F_2= F_1 \oplus F_0=10$
n=3	$f_3= f_2+ f_1$	$F_3= F_2 \oplus F_1=101$
n=4	$f_4= f_3+ f_2$	$F_4= F_3 \oplus F_2=10110$
n=5	$f_5= f_4+ f_3$	$F_5= F_4 \oplus F_3=10110101$
n=6	$f_6= f_5+ f_4$	$F_6= F_5 \oplus F_4=1011010110110$
⋮	⋮	⋮
n=k	$f_k= f_{k-1}+ f_{k-2}$	$F_k = F_{k-1} \oplus F_{k-2}$

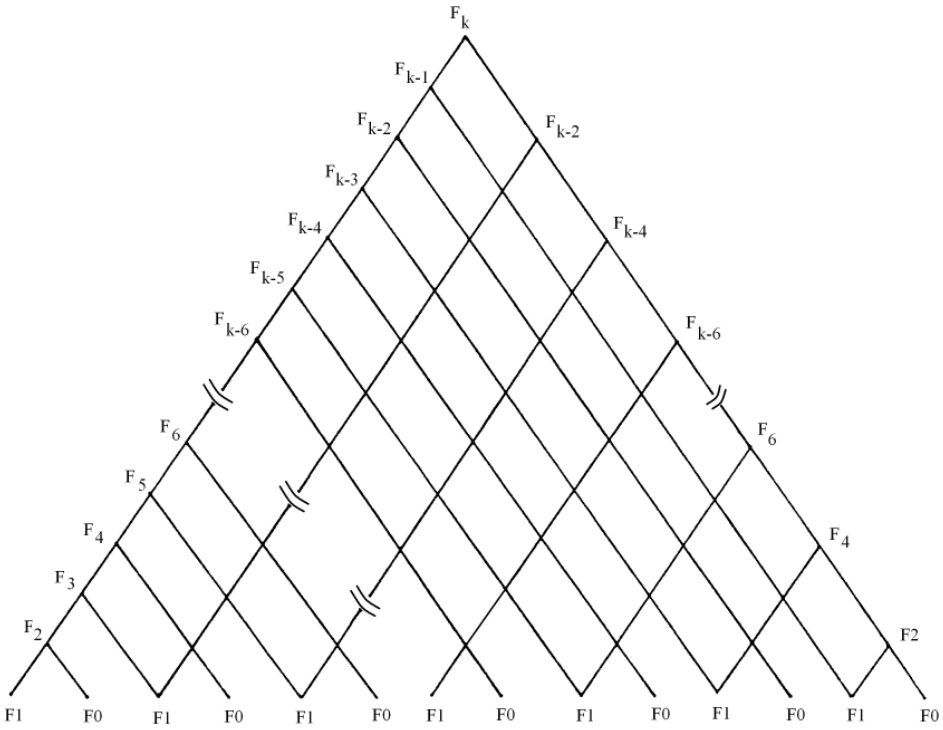


Figure 8.1 Fibonacci Blocks configuration in terms of unit blocks.

Lemma The LZ complexity of the third Fibonacci block: $S = F_2 = 10$ is two.

Proof:

Step 1) Insert first bit $s_1 = 1$; then the reconstructed sequence becomes $S' = 1\backslash$

Step 2) The second bit is $s_2 = 0$; the current subsequence will be $Q = s_2 = 0$; then the concatenated subsequence is $S'Q = 10$, and the previous subsequence is $S'Q\hat{} = 1$. The vocabulary of the previous subsequence is $V(S'Q\hat{}) = \{1\}$. Since $(Q = 1) \in V(S'Q\hat{})$, i.e., the dictionary contains Q. Thus Q is not a new subsequence. Hence $S = S' = s_1 \backslash s_2 = 1\backslash 0$ has two segments, thus $C(S) = C(F_2) = 2$.

Theorem 1 $S'' \underset{B}{\subset} S$ denotes a Fibonacci block S with length r and a sub block of S, say S'' with length *(n-r).* The Fibonacci block S' is constructed from concatenation of sequences S and S''

or $S' = S \oplus S''$. The LZ complexity of $S' = S \oplus S''$ is $C(S') = C(S) + 1$, where $C(S)$ is the LZ complexity of block S.

Proof:

$$S' = S \oplus S'', \text{ where } S'' \underset{B}{\subset} S,$$

$$L(S') = L(S \oplus S'') = n,$$

$$S' = s_1 s_2 \cdots s_r s_{r+1} \cdots s_n,$$

$$S'' = s_{r+1} \cdots s_n.$$

$S = s_1 s_2 \cdots s_r$ has complexity $C(S)$. We start the complexity analysis of S' by reading element $(r+1)$ from the buffer.

Step r + 1 The $(r+1)^{th}$ character is s_{r+1}. The reconstructed sequence is $S' = S \setminus = s_1 s_2 \cdots s_r \setminus$, the current subsequence is $Q = s_{r+1}$, the concatenated subsequence $SQ = s_1 s_2 \cdots s_r s_{r+1}$, and the previous subsequence is $SQ\wedge = s_1 s_2 \cdots s_r$. Since $S'' \underset{B}{\subset} S$ then $(Q = s_{r+1}) \in S$, hence $Q \in V(SQ\wedge)$; this means the dictionary contains Q, so it is not a new subsequence.

Step r + 2 The $(r+2)^{th}$ character is s_{r+2}. The reconstructed sequence is $S' = S \setminus = s_1 s_2 \cdots s_r \setminus$, the current subsequence is $Q = s_{r+1} s_{r+2}$, the concatenated subsequence $SQ = s_1 s_2 \cdots s_r s_{r+1} s_{r+2}$, and the previous subsequence is $SQ\wedge = s_1 s_2 \cdots s_r s_{r+1}$, since $S'' \underset{B}{\subset} S$, then $(Q = s_{r+1} s_{r+2}) \in S$, hence $Q \in V(SQ\wedge)$, this means the dictionary contains Q and it is not a new subsequence.

This process will be repeated up to the last term. For last term we write:

Step n The last n^{th} character is s_n. The reconstructed sequence has not been changed and still is $S' = S \setminus = s_1 s_2 \cdots s_r \setminus$, the current subsequence is $Q = s_{r+1} s_{r+2} \cdots s_n$, the concatenated

subsequence is $SQ = s_1 s_2 \cdots s_r s_{r+1} s_{r+2} \cdots s_n$, and the previous subsequence is $SQ^{\wedge} = s_1 s_2 \cdots s_r s_{r+1} s_{r+2} \cdots s_{n-1}$. Since $S'' \underset{B}{\subset} S$ then $\left(Q = s_{r+1} s_{r+2} \cdots s_n\right) \underset{B}{\subset} S$, hence $Q \in V(SQ^{\wedge})$, this means the dictionary contains Q and it is not a new subsequence. Since $Q = s_{r+1} s_{r+2} \cdots s_n = S''$ then the final reconstructed sequence is $S' = S \setminus Q = S \setminus S''$, this means concatenating the sequence S and a subsequence S'' generates one more partition in the reconstructed sequence, which in turn increases LZ complexity by one unit. Consequently the LZ complexity of S' is: $C(S') = C(S \oplus S'') = C(S \setminus S'') = C(S) + 1$.

Theorem 2 The LZ complexity of a Fibonacci block, F_n is equal to *n.*

The proof is straightforward using previous lemma and Theorem 1.

$$C(F_n) = C(F_{n-1}) + 1 = C(F_{n-2}) + 2 = C(F_{n-3}) + 3 = \cdots = C(F_2) + (n-2) = 2 + (n-2) = n.$$

Theorem 3 The LZ complexity of a Fibonacci blocks sum is $n + 1$.

$$\sum_{j \oplus = 1}^{n} F_j = F_n \oplus F_{n-1} \oplus F_{n-2} \oplus F_{n-3} \oplus \cdots \oplus F_2 \ .$$

Proof The proof is straightforward using theorem 1.

$$H(n) = \sum_{j=1 \oplus}^{n} \left(F_j\right) = F_n \oplus F_{n-1} \oplus F_{n-2} \oplus F_{n-3} \oplus \cdots \oplus F_2 \ ,$$

$$C\left\{ \sum_{j=1 \oplus}^{n} \left(F_j\right) \right\} = C\left\{F_n \oplus F_{n-1} \oplus F_{n-2} \oplus F_{n-3} \oplus \cdots \oplus F_2\right\} = n+1.$$

The following theorem is the general form of the previous theorem.

Theorem 4 The LZ complexity of Fibonacci blocks sum

$G(n) = F_n \oplus \cdots \oplus F_m \oplus \cdots \oplus F_l \oplus \cdots \oplus F_k$,

where $k < l < m < n$, is equal to $n+1$.

Proof The proof is straightforward using Theorem 1.

$$C\{G(n)\} = C\{F_n \oplus \cdots \oplus F_m \oplus \cdots \oplus F_l \oplus \cdots \oplus F_k\} = C\{F_n\} + 1 = n + 1,$$
$$\because F_k \subset F_l \subset F_m \subset F_n.$$

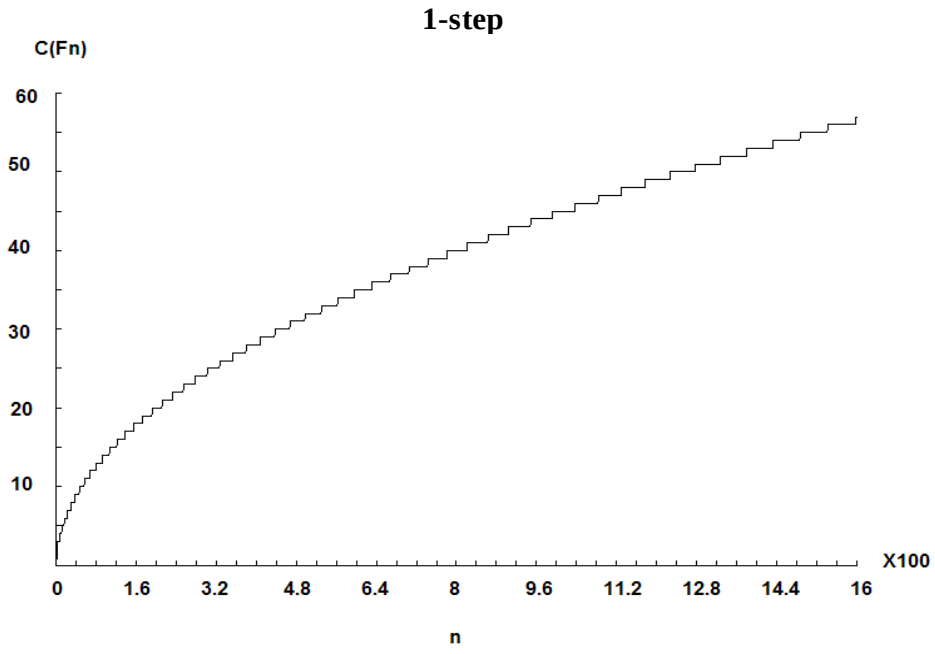
Theorem 5 The LZ complexity of the family of evolutional Fibonacci-like blocks:

$F_{n+m+2} = F_{n+m+1} \oplus F_n$, where $m < n$ is equal to $n + m$.

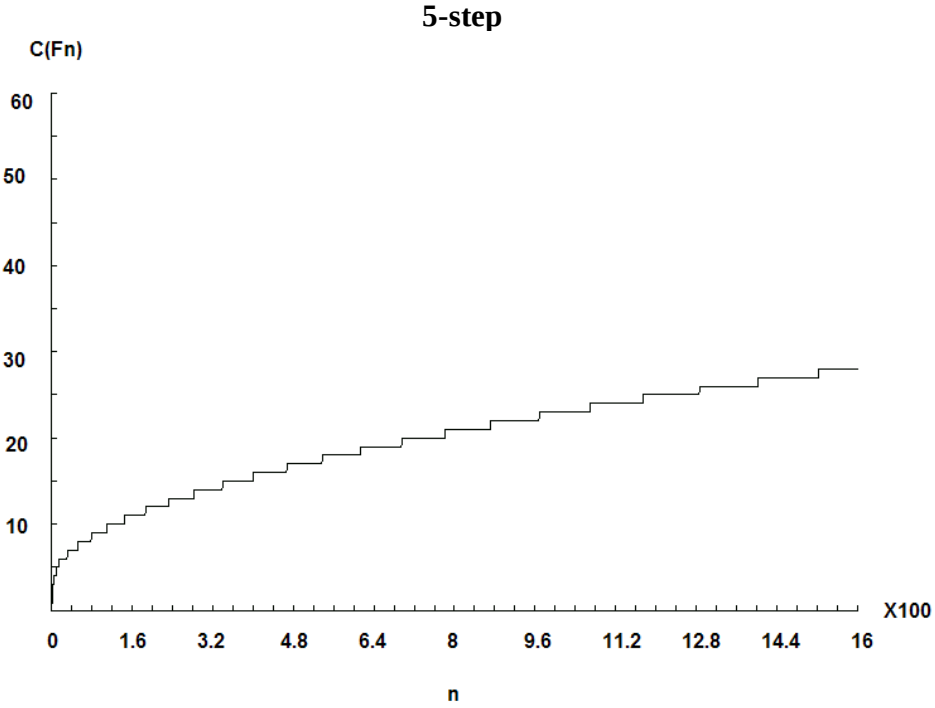
Proof

$$F_{n+m+2} = F_{n+m+1} \oplus F_n,$$
$$m = 0: F_{n+2} = F_{n+1} \oplus F_n,$$
$$m = 1: F_{n+3} = F_{n+2} \oplus F_n,$$
$$m = 2: F_{n+4} = F_{n+3} \oplus F_n,$$
$$\vdots$$
$$m = k: F_{n+k} = F_{n+k-1} \oplus F_n,$$
$$C\{F_{n+k}\} = C\{F_{n+k-1} \oplus F_n\},$$
$$\because F_n \subset F_{n+k-1},$$
$$C\{F_{n+k}\} = C\{F_{n+k-1}\} + 1 = C\{F_{n+k-2}\} + 2 = \cdots = C\{F_n\} + k = n + k = n + m.$$

A computer simulation for different correlation lengths is shown below. One can see that increasing the correlation length causes the slope of the exponential growth to decrease.

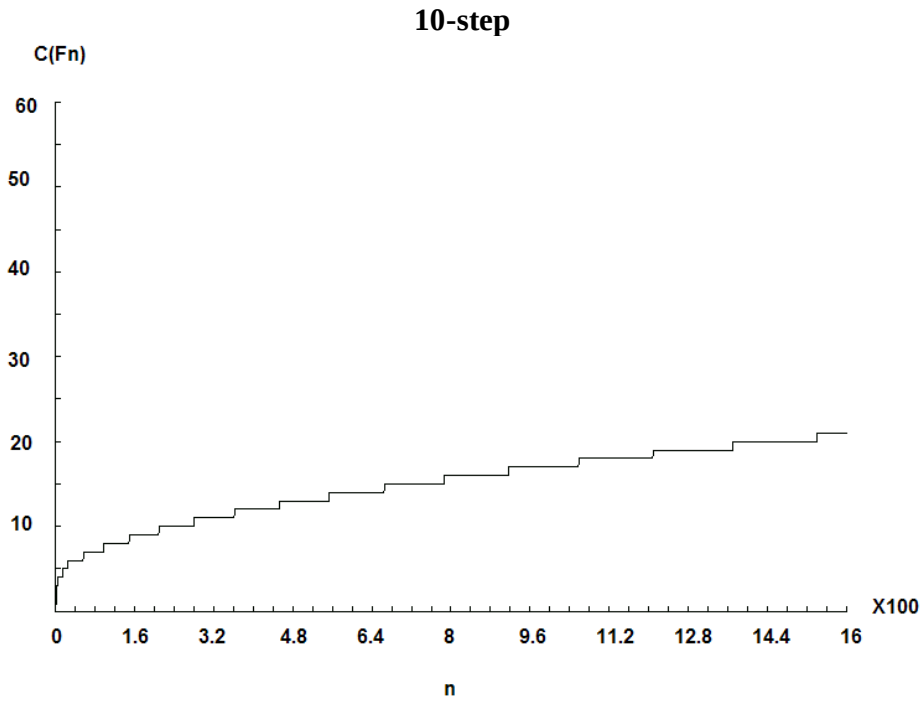


(a)

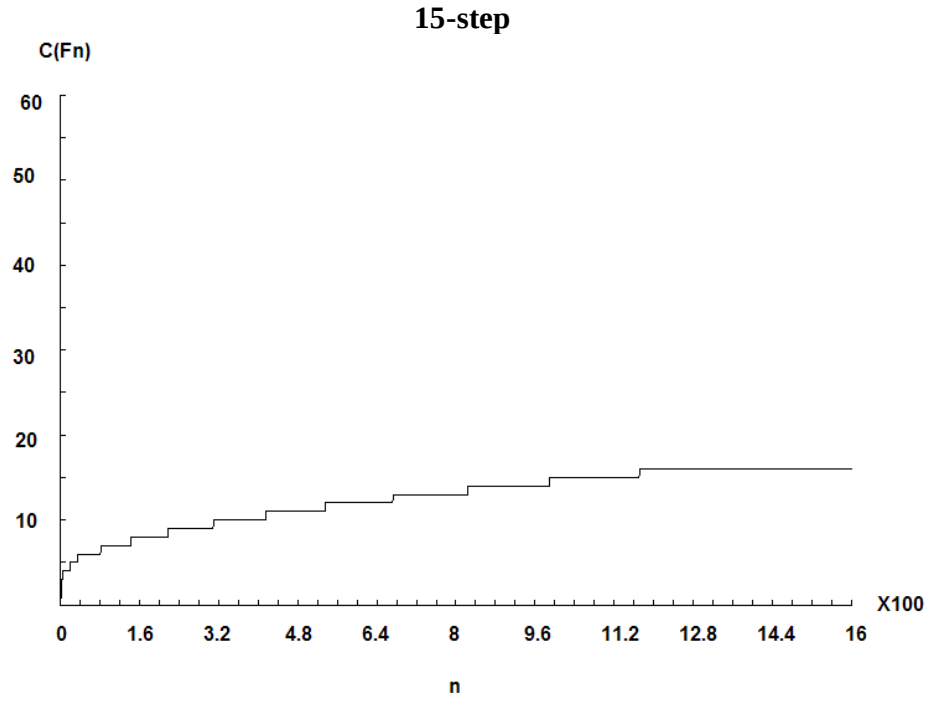


(b)

Figure 8.2 Plot of Fibonacci map LZC vs. string length *n* for (a) 1-step (b) 5-step correlation length.

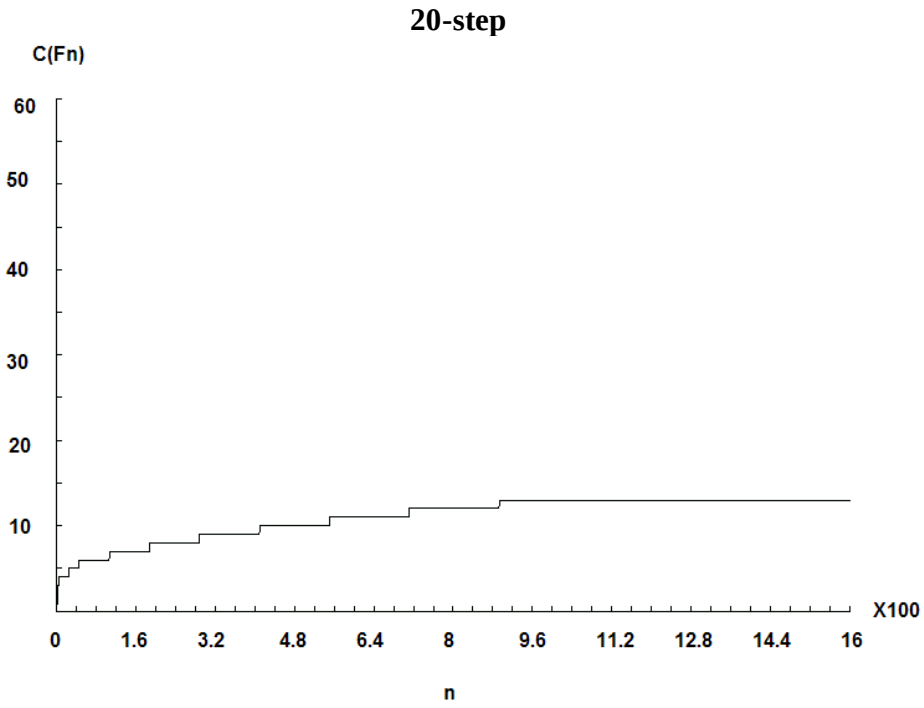


(c)



(d)

Figure 8.2, cont. Plot of Fibonacci map LZC vs. string length *n* for (c) 10-step (d) 15-step correlation.



(e)

Figure 8.2, cont. Plot of Fibonacci map LZC vs. string length *n* for (e) 20-step correlation.

8.2 Fibonacci System LZ Complexity and Lyapunov Exponent

We show analytically that the algorithmic complexity of a block of a binary Fibonacci sequence increases logarithmically with the block length. This means $C(F_n) = \log_2(f_n)$ where f_n is the block length of the Fibonacci generator map defined by relation:

$$f_n = \left(\frac{\tau^n - \left(-\frac{1}{\tau}\right)^n}{\sqrt{5}} \right).$$

where τ satisfies the equation $\tau - \left(\frac{1}{\tau}\right) = 1$, and its solutions are

$$\alpha, \beta = \frac{\left(1 \pm \sqrt{5}\right)}{2}.$$

Then

$$C(F_n) = \log_2(f_n) = \log_2\left(\tau^n - \left(-\frac{1}{\tau}\right)^n\right)\left(\frac{1}{\sqrt{5}}\right) = \log_\tau\left\{\left(\tau^n - \left(-\frac{1}{\tau}\right)^n\right)\left(\frac{1}{\sqrt{5}}\right)\right\}(\log_2\tau)\,.$$

Since $\left|\frac{1}{\tau}\right| < 1$, when n is large we can write, $f_n \approx \left(\frac{\tau^n}{\sqrt{5}}\right)$, thus

$$\lim_{n\to\infty} C(F_n) = \lim_{n\to\infty} \log_2(f_n) = \lim_{n\to\infty} \log_\tau\left(\frac{\tau^n}{\sqrt{5}}\right)(\log_2\tau) \cong n.$$

We have examined the block patterns by calculating the LZ complexity. Figure 8.3 shows the plot of LZ complexity vs. Fibonacci block length f_n. The complexity measure is increased by one unit from one block to the next the one. This is expected from first lemma. This example shows the existence of long-range correlation in this evolutionary structure which is constructed from less complex units. Figure 8.4 illustrates the plot of Fibonacci map LZ complexity vs. block length when n is large. Notice that the normalized LZ complexity of a quasi-periodic signal asymptotically approaches zero for long data bits.

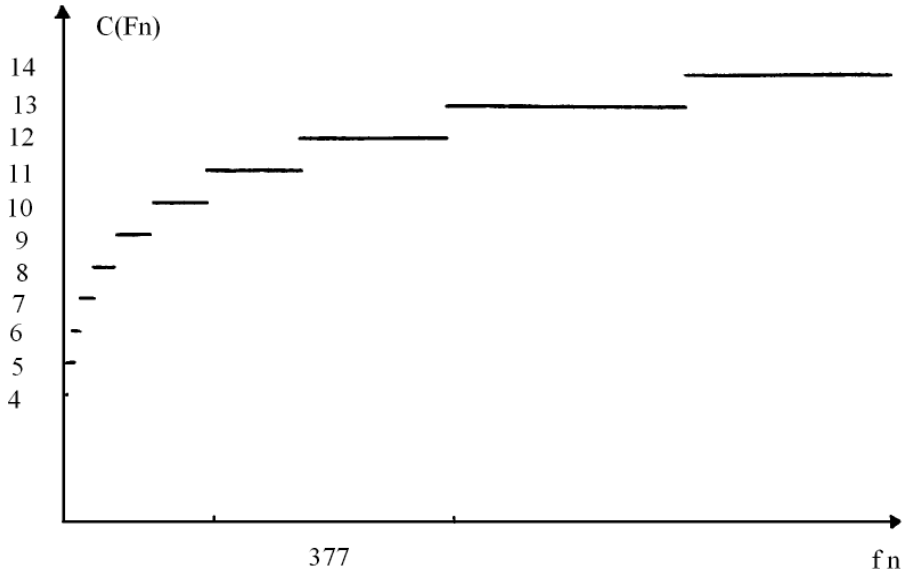


Figure 8.3 Plot of Fibonacci map LZC vs. block length f_n.

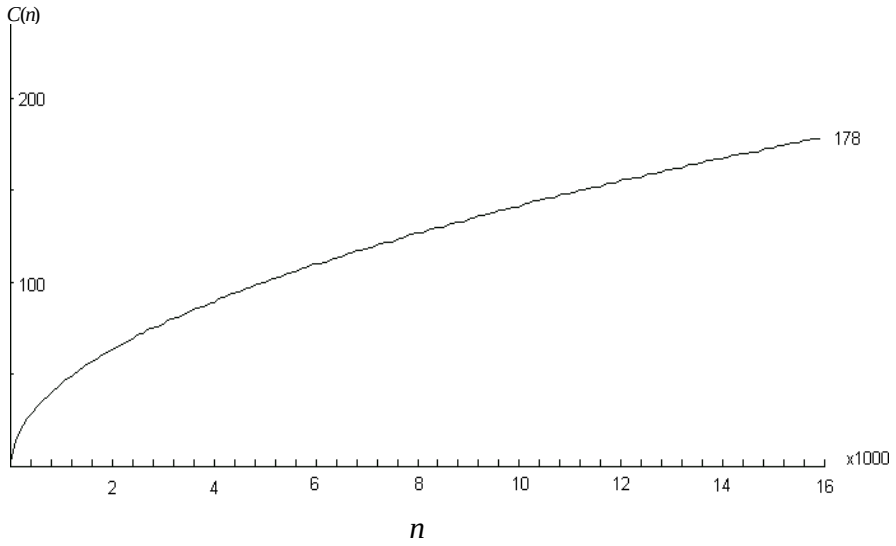


Figure 8.4 Plot of Fibonacci map LZC vs. string length *n*.

Next we determine the relation between the LZ complexity measure and the Lyapunov exponent for a Fibonacci map:

$$F:\begin{cases} X_{n+1} = X_n + Y_n \\ Y_{n+1} = X_n \ . \end{cases}$$

By use of Theorem 1 the LZ complexity for each block is calculated by,

$$C_b = \lim_{n\to\infty} \frac{C(F_n)}{n} = \lim_{n\to\infty}\left(\frac{n}{n}\right) = 1 \quad \text{Block/Iteration.}$$

In order to compute the Lyapunov exponent for a Fibonacci system, we need to solve the variational equation for this dynamic system. Then the variational equation, $\vec{Z}_{n+1} = \overline{\overline{J}}\Big|_{\vec{X}_n} . \vec{Z}_n$ has initial unit vector $\vec{Z}_0$. The solution of the variational equation at step *n* will be

$$\vec{Z}_{n+1} = \overline{\overline{J}}\Big|_{\vec{X}_n} . \vec{Z}_n = \overline{\overline{J}}\Big|_{\vec{X}_n} . \overline{\overline{J}}\Big|_{\vec{X}_{n-1}} . \overline{\overline{J}}\Big|_{\vec{X}_{n-2}} \ldots \overline{\overline{J}}\Big|_{\vec{X}_1} . \overline{\overline{J}}\Big|_{\vec{X}_0} . \vec{Z}_0$$

$\overline{\overline{J}}\Big|_{\vec{X}_n}, \ldots, \overline{\overline{J}}\Big|_{\vec{X}_1}, \overline{\overline{J}}\Big|_{\vec{X}_0}$ are Jacobean matrices determined for each point like $\vec{X}_m$. Since the Jacobian

matrices of a Fibonacci system are all constant and equal to

$$J_n = J_{n-1} = J_{n-2} = \cdots = J_1 = J_0 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \equiv J,$$

then the solution of variational equation at step *n* will be:

$$\vec{Z}_n = J^n . \vec{Z}_0 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \vec{Z}_0 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} f_1 \\ f_0 \end{pmatrix} = \begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix}.$$

Then the Lyapunov exponent is

$$\lambda = \lim_{n\to\infty}\left(\frac{1}{n}\right)\log_\tau\left(\frac{\left\|\vec{Z}_n\right\|}{\left\|\vec{Z}_0\right\|}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right)\log_\tau\left(\frac{\left\|(f_n, f_{n-1})\right\|}{\left\|(f_1, f_0)\right\|}\right),$$

$$= \lim_{n\to\infty}\left(\frac{1}{2n}\right)\log_\tau\left(\frac{f_n^2 + f_{n-1}^2}{f_1^2 + f_0^2}\right),$$

$$= \lim_{n\to\infty}\left(\frac{1}{2n}\right)\log_\tau\left(f_n^2 + f_{n-1}^2\right) = 1 \ \textit{Block / iteration}\ .$$

This means a Fibonacci system generates information with the rate of one block per iteration. On the other hand, as the LZ complexity measure suggests, the information capacity increase rate is one block per iteration, or C_b=1. Interestingly, for a Fibonacci system the Lyapunov exponent and LZ complexity are equal. Also, this result addresses one of the most acknowledged features of a quasi-organized system and quantitatively locates the quasi-organized systems between highly organized with very low complexity, and constrained-unorganized systems, that is, the chaotic systems with high complexity measure. This analysis provides some evidence that we can make a definite conclusion about long-range evolutionary strings. These systems are neither short-range periodic nor chaotic or random. Therefore, for large data strings we propose the following conclusion for the normalized complexity measure, for periodic, quasi-periodic, chaotic and random data respectively.

$$\lim_{n\to\infty}\left\{\frac{K}{b(n)}\right\}=\lim_{n\to\infty}\left\{\frac{K\log_2 n}{n}\right\}=0\text{ , where } K \text{ = Constant.}$$

$$\lim_{n\to\infty}\left\{\frac{C(n)}{b(n)}\right\}=\lim_{n\to\infty}\left\{\frac{(\log_2 n)^2}{n}\right\}=0,$$

$$0<\lim_{n\to\infty}\left\{\frac{C(n)}{b(n)}\right\}<1,$$

$$\lim_{n\to\infty}\left\{\frac{C(n)}{b(n)}\right\}=1.$$

We confirm numerically the asymptotic behavior of the normalized LZ complexity versus the length of the array. The results are summarized in Figure 8.5. We conclude that using a normalized algorithmic complexity measure, as a pattern classifier, gives a non-zero value less than one for chaotic responses. For periodic and quasi-periodic responses, as the data string grows the complexity approaches zero, with a higher deceasing rate is observed for periodic responses.

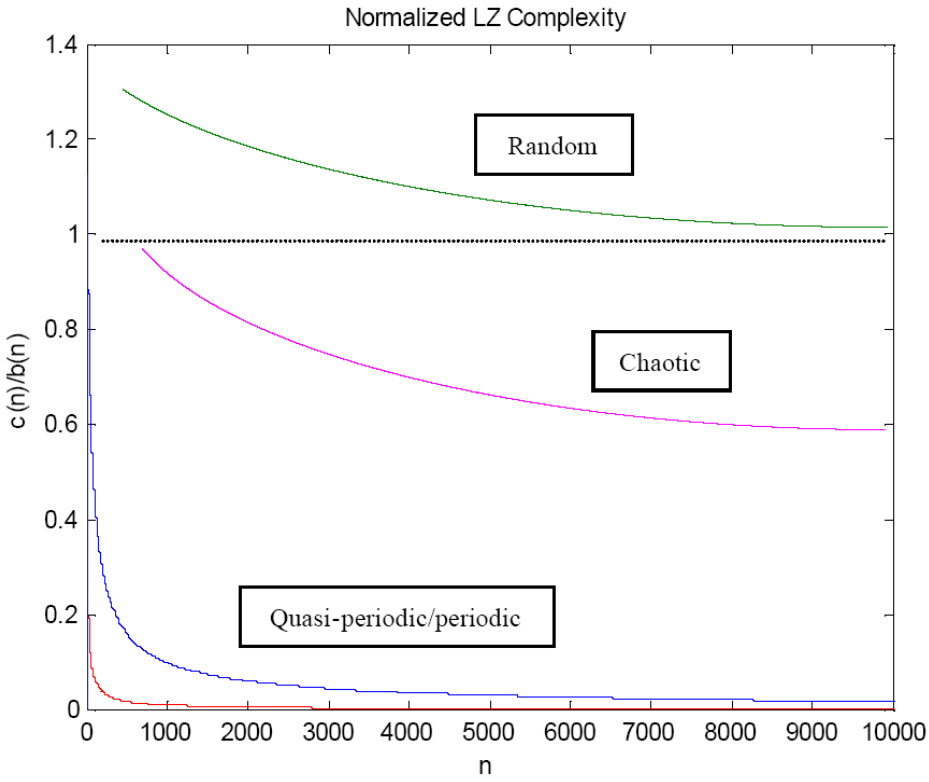


Figure 8.5 Comparison of asymptotic normalized LZC.

Theorem 6 The LZ complexity of the sequence $U(n)$, which can be expanded in terms of the basis units of Fibonacci blocks as a unit vector $U(n)=\sum_{k=1\oplus}^{M} a_k F_k$, is equal to

$$C\{U(n)\}=C\left\{\sum_{k=1\oplus}^{M} a_k F_k\right\}=M+1.$$

The significance of these results is in the study of complexity analysis of strings, or of objects like biomolecules, species, documents and programs which can be mapped to strings of alphabets. Throughout this chapter, the main aim of the investigation is the analysis of strings carrying information generated by a quasi-periodic system. Our special interest is whether or not long range structures exist in such sequences. This analysis gives some justification for the fact that information-carrying strings are characterized by correlations which are ranging from the

beginning of the string up to the end of the string. For example, after listening to the beginning of a Bach Preludium, where the general theme is worked out, we expect to hear the theme again in many variations up to the very end. Such expectations are justified only if there exist indeed long-range correlations. This is the scientific expression of our intuitive expectations. Long range correlations provide a strong selection criterion. Another reason to expect long range correlations is that evolution probably operates in regions of self-organized criticality. As known from many examples in the natural sciences, critical conditions (i.e., operating near to transitions) imply the existence of long range correlations. The extension of this work to higher alphabet size will adapt the algorithmic complexity approach to multi-dimensional data strings. We expect higher dimensional algorithmic complexity measures can be straightforwardly adapted to become tolerant to measurement errors as they occur, for example, in biological data.

8.3 Conclusion

We have demonstrated a computational metric to characterize and classify signals based on their distinguishing patterns. We have investigated the whole set of control parameter values that generate chaotic attractors with a new exploration tool. To characterize their responses for different values of control parameters we have computed the LZ algorithmic complexity measure of phase space attractors. We showed the usefulness of the LZ complexity measure as a metric to characterize the patterns of discrete and continuous dynamical systems with chaotic and periodic responses. This is important because it provides a computational metric to find and classify complex patterns. Our efforts here have been directed toward outlining a computational algorithm and giving example of the complexity measure for the Fibonacci map as a quasi-periodic system. In previous chapters, we have observed that the LZ-complexity of chaotic responses is much higher than periodic responses. In addition we found that for quasi-organized

systems like the Fibonacci map, complexity grows logarithmically with the length of the data block. That is, the order of organization for this quasi-organized system is increasing logarithmically with the evolutionary length of the system. In general, normalized algorithmic complexity turns out to be a non-zero value less than one for chaotic responses and zero for periodic and quasi-periodic responses for large strings. This work extends the usefulness of the complexity measure as a metric to characterize the dynamic patterns of discrete and continuous systems with chaotic, periodic, and quasi-periodic responses.

CHAPTER 9

COMPLEXITY ANALYSIS OF CONVOLUTIONAL CODING WITH CHAOTIC ENCRYPTED DATA

9.1 Design of Chaotic Cryptosystems

The main purpose of information security is to provide confidentiality, integrity, authentication, and non-repudiation. Confidentiality is the goal of preventing intentional or unintentional disclosure of communication between a sender and a recipient. Integrity is the goal of ensuring the accuracy and consistency of information during all processing (storage, transmission, and so forth). Authentication is the process of determining the identity of an authorized user through matching a user name and password, a fingerprint or retinal scan, a smart card and PIN, and so on. Non-repudiation is ensuring that the sender can't deny sending a message and the receiver can't deny receiving it.

Encryption is the process of concealing information to render it unreadable except for the intended recipients. To provide confidentiality, secure messages can be communicated via an encryption algorithm. Encryption algorithms are used in many areas such as access control protocols, shared control schemes, authenticated distributed computing, e.g., virtual private network, voting systems, electronic money, cyber risk insurance and distributed management of data bases.

Secure encryption is a fundamental problem in the field of cryptography. An encryption system is called secure if knowing the encrypted message does not give any partial information about message that is not known beforehand. The idea of applying a chaotic systems algorithm to

design cryptosystems, data encoders and chaotic spread spectrum modulation schemes originates from the intrinsic characteristics of chaotic systems. These characteristics are complex output responses and sensitive dependence on initial conditions which makes the system state highly unpredictable and its output response random looking. Therefore, in its nature, chaos is a good candidate for pseudo-random number generators and encryption algorithms; this is due to its major characteristic of sensitively depending on its initial condition and mixing property. According to Shannon, in a good mixing transformation, functions should be complicated, involving all variables in a sensitive way [113]. Owing to the high complexity and nonlinearity of signals generated by a chaos generator, when it is imposed onto a data-stream, the output signal stream will become highly irregular and seems to be totally unpredictable which increase the difficulty for intruders to comprehend these data streams. Thus, chaos signal generators are commonly adopted as so-called signal masks for data encryption. For a typical chaotic system, even when there is a very minor variation in the initial conditions, the system dynamics will be varied tremendously. Based on this concept, when a chaotic data generator is used as the key(s) in a cryptosystem, sensitivity to initial conditions will substantially increases the complexity for a hacker to guess and hack the system. In particular, if a hyperchaotic system output is applied, higher complexity can be generated. The basic idea here is to choose nonlinear systems satisfying the mixing property, with robust chaos and with large control parameter sets. The mixing property of chaotic functions is closely related to the property of diffusion in encryption algorithm. Diffusion means spreading out of the influence of a single plaintext digit over many cipher-text digits so as to hide the statistical structure of the plain-text. Robust chaotic systems can eventually ensure the diffusion property in the key space. On the other hand, non-robust systems may have weak encryption keys. Consequently, we need to consider only those systems

that have robust chaos for a large span of control parameters. This provides a large verity of encryption keys. These suggestions lead to the consideration of many systems that are studied in previous chapters. Among those, well-known logistic map and 4-dim feedback controlled system are chosen for our study.

Mainly two approaches to chaotic ciphers are known. One is a cryptosystem based on a chaos synchronization technique of analog circuits. The second is the chaos-based cipher realized on digital circuits or computers with a finite precision effect. In the past, most experiments with chaotic cryptosystems were carried out between two circuit boards which there were connected by an ideal secure channel. This approach is not suitable for real-time applications like the Internet where packet loss during data transmission is commonly found. The traditional key value of the chaotic equation is a floating point number. This means that the key length is determined by the precision value of a floating point number. The computational complexity of most chaotic cryptosystems suffers most from the complex floating point arithmetic calculations which in turn affect the overall performance and efficiency of the cryptosystems, especially in real-time design. In addition, there are two intrinsic problems for setting the key length in chaotic cryptosystems. One is that most chaotic systems are known and within these only a relatively narrow range of options to be chosen from which to choose are available. Secondly, some of the nonlinear equations being proposed in the literature have a narrow range of control parameters to be used as a system key with chaotic behavior. However, as we have presented and discussed in previous chapters, these problems can be resolved by introducing new designs of chaotic systems. In designing chaotic systems for data encryption two factors are important. One factor is encryption-decryption overall time and the second factor is the level of security. The computation time for encryption and decryption depends on the

complexity of the designed system and the value of state variables. Hence, the lower the complexity of the equations, the shorter the computation time will be. On the other hand, if the complexity of equation is high, a longer time will be needed for data encryption and decryption. In order to choose an equation with lower complexity, a discrete chaotic map is suggested. If the chaotic equations are discrete maps, encryption algorithm only involve basic arithmetic operations like summations, subtractions, multiplications and divisions. On the other hand, if the nature of the chaotic equation is a continuous flow, encryption and decryption involve differentiation or integration type operations when calculating the value of next state variable. In either case, from a data complexity point of view, a binary value of a state variable is more preferable. If the value of state variable is a binary sequence, it takes a shorter time for computing the value of the next state variable. On the other hand, if the value of the state variable is a floating point number, it requires a longer time for computing the value of the next state variable. The main theme of this chapter is the evaluation of a new design that utilizes a chaotic system and a convolutional encoder which uses binary state variables in its responses. But before introducing our design we need to address the central problem in any chaotic encoding technique, that is, how to encode an arbitrary message into the waveform generated by the chaotic oscillator using symbolic dynamics. Throughout previous chapters of this dissertation we have presented the possibility of utilizing symbolic representations of chaotic orbits for encoding with chaotically behaving signals. The success of this type of nonlinear digital encoding scheme relies on partitioning the phase space properly so that a good symbolic dynamics can be defined. The implementation of this approach is based on the following idea. Imagine a chaotic oscillator that generates a signal consisting of an apparently random sequence of positive and negative peaks. A possible way to assign a symbolic representation to the signal

is to associate a positive peak with a one, and a negative peak with a zero, thereby generating a binary sequence. One advantage of this type of communication strategy is that the nonlinear chaotic oscillator that generates the wave form for transmission can remain simple and efficient, while all the necessary electronics controlling encoding of the signal remain at a low-powered microelectronic level.

Throughout this chapter we address the following questions. How much can information complexity be increased via a chaotic encryption and convolutional coding? What types of chaotic systems or dynamical invariant sets are suitable for a general encrypting scheme based on their characteristic performance? And what is the influence of noise and data size on this scheme. The answers to these questions constitute an essential step in the development of a practical framework in designing secure and dependable system for digital communication.

9.2 Chaotic Cryptosystems Synchronization

Most chaotic encryption methods are basically symmetric key encryption in which both the encryption and decryption keys being used are the same set of chaotic equations. In most of the cases, the parameters of these chaotic equations and their initial values of the state variable are used as the encryption keys (the symmetric keys). Hence, the level of security will depend on two primitive factors: the key length and the output of the encrypted cipher. If the key length or numbers of keys are small, it shortens the time of cryptanalysis of the keys. A large number of sets of chaotic equations induce complexity in cryptanalysis to provide a better security level and a small set of chaotic equations makes it easier for cryptanalysis.

There are two main methods to perform chaotic encryption. One is by a synchronization technique and the other one is by a self-synchronization method. Chaotic real-time encryption based on a synchronization technique uses two identical or non-identical sets of nonlinear

chaotic systems. At the transmitter, the chaotic equation generates chaotic signal. It then uses an add-up function to mask the original signal with the chaotic signal. At the receiver, the chaotic equation generates a chaotic signal which is same as that in the transmitter side. It then uses a reverse masking function to retrieve the original signal using the received signal and the generated chaotic signal (Figure 9.1). Based on this technique, data can be transmitted using a non-guarantee protocol by means of the mechanism of transmitting the number of packets with the data packets. As a result, if a data packet is lost, synchronizing the chaotic signals on both sides is based on the number of packets. One disadvantage of this technique is that the key must be shared in the secure channel.

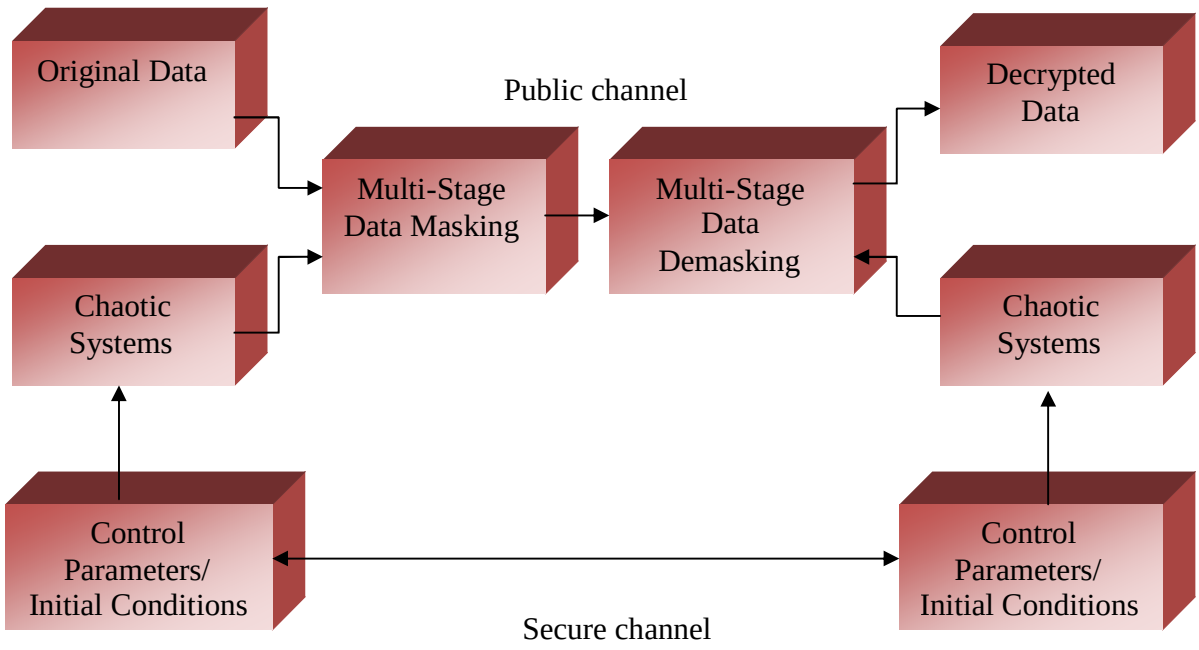


Figure 9.1 Synchronization in a chaotic cryptosystem.

In a self-synchronization technique, chaotic real-time encryption is accomplished using two identical sets of continuous chaotic equations such as coupled synchronized systems. The receiver receives a driven message from the sender. At the transmitter, the original signal is imposed on the chaotic equation and is output as a chaotic signal which will be transmitted to the receiver. At the receiver, the received chaotic signal will be injected into the chaotic equation and output as the original signal. The iteration time is also transmitted with the data packet. It can be used as a key for both discrete and continuous systems. Under this chaotic real-time encryption scheme, a certain time is needed to synchronize system state variables with the transmitter (Figure 9.2). From an implementation point of view, this real-time encryption scheme may not be suitable for most online real-time streaming services. The main reason is that it is possible to lose Real-time Transport Protocol (RTP) packets. RTP is an application layer transport protocol providing essential support for multimedia streaming and distributed computing along with the Real Time Streaming Protocol (RTSP) [114]. If the receiver misses some RTP packets, self-synchronization cannot be achieved because in order to synchronize the state variables, of both sides the receiver needs to have the previous state variables. The advantage of this scheme is that the key can be shared in the public channel and its disadvantage is that the data must be transmitted in a guaranteed protocol.

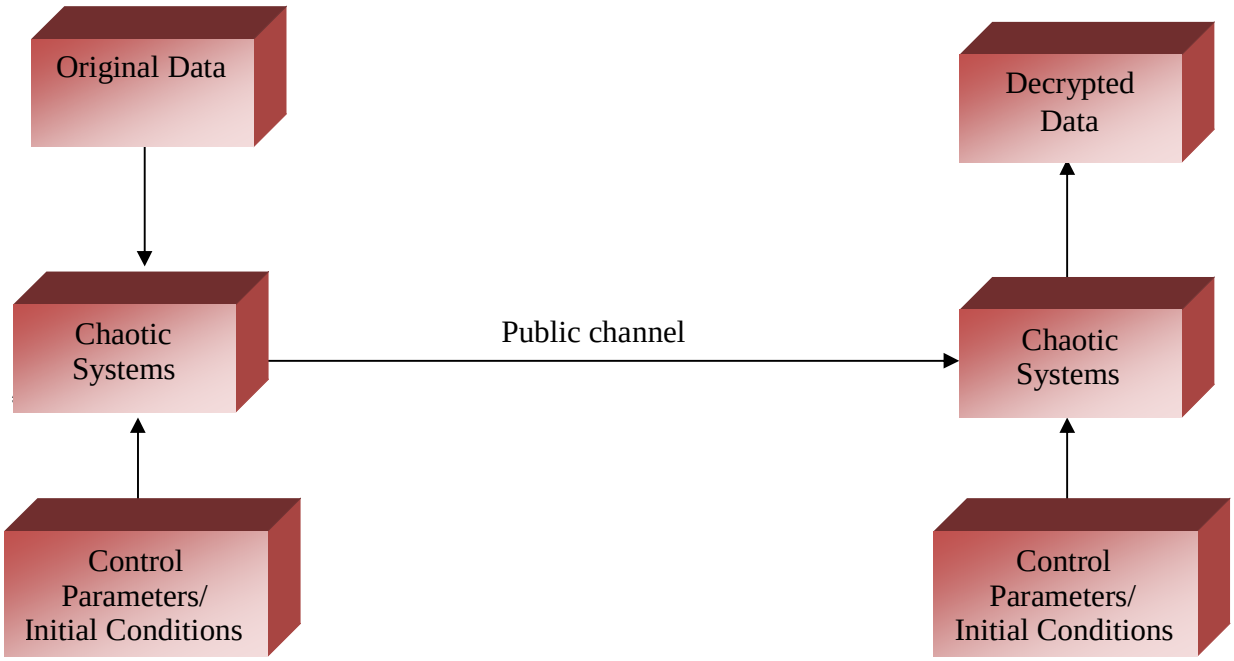


Figure 9.2 Self-Synchronization in a chaotic cryptosystem.

9.3 Chaotic Encryption and Convolutional Coding

In this section, chaotic encryption with a convolutional encoding algorithm is introduced as a new approach in the design of crypto-coded system with error correcting capability. Then, the characteristics of various convolutional encoders with chaotic states which are observed in a logistic map and a chaotic and hyperchaotic feedback controlled oscillator are studied. We compare the performance of various encoding algorithms. We present numerical evidence that indicate the algorithmic complexity associated with a chaotic system has a non-decreasing form. This behavior coincides with the free distance (d_{free}) behavior for the applied class of codes, which is an indicator of better error-correcting strength. One of the recent applications of convolutional encoders is reported in a digital fingerprinting technique for the merchant who can embed unique buyer identity marks into digital media copy, and make it possible to identify

traitors who redistribute their illegal copies. Authors introduce the definition of the Fingerprinting Information Code and a practical construction method accomplished by the composing of convolutional codes and using fingerprinting codes based on Boneh-Shaw model [115, 116]. They provide an approach for encoding and decoding fingerprints for the user's private information based on a convolutional code with an optional code set for Intellectual Property Protection [117].

In our framework, chaotic encryption of the message signal and convolutional encoding of the enciphered message have been incorporated. This is done by combining the output of a nonlinear system with a convolutional encoding algorithm. By analyzing the chaos region, a set of different control parameters and output responses are found to be suitable for being the keys of encryption. Then, the performance is evaluated. The performance of the convolutional encoders is measured by computing their bit error rates versus the channel noise and the performance of the combined system is evaluated by the LZ complexity measure. The proposed schemes have very little loss in bit-error rate performance so that they are a good alternative to the design of systems for which encryption and error correction are important joint goals. The schematic block diagram of the proposed system is depicted in Figure 9.3. It should be noted that we have based our description on the simplest case of a single map key-generator. While this produces a secure key, the simplicity of the key might make it vulnerable to a sophisticated attack. In general, the keys used for encryption can be the parameters of the chaotic maps and initial conditions of the variables since chaotic sequences show high sensitivity to both parameters. The message is encrypted using a chaotic sequence with a control parameter as key 1 and an initial condition as key 2. Then the encrypted message is encoded using a rate-*k/n* error correcting code as key 3, producing the encoded message. This permits reliable communication

and ensures the integrity of the data of an information sequence over a channel that adds noise, introduces bit errors, or otherwise distorts the transmitted signal. For further secrecy, the resulting embedded signal can then interleaved using key 4 and so on. The resulting message is then transmitted to the receiver. At the receiver end, the message is received and the hidden message signal is extracted by doing the reverse process using the same keys as those used during encoding. Since the encrypted message signal is encoded using a convolutional encoding process, a filtering operation is done at the receiver to produce an estimate of the encrypted message. When an estimate of the interleaved embedded signal is produced, this signal is deinterleaved using key 3 to produce an estimate of the embedded signal. By use of key 2, the chaotic sequence is regenerated, the embedded signal estimate is then decoded, and an estimate of the encrypted message is produced. The error control decoder then decodes the estimate of the encrypted message, which is decrypted using key 1 to recover the original message. The system consists of a transmitter module and a receiver module. The transmitter module consists of a chaotic system and a digital encryption mechanism. First, an equivalent digital key sequence is generated from one of the chaotic signals by a suitable threshold mechanism. This could be done by assigning the states 1 or 0 depending on whether the chaotic signal amplitude is above or below a certain threshold level. Then this chaotic digital key is Exclusively-ORed (XOR) with the digital information signal to generate the encrypted signal. This chaotic signal is also transmitted through the channel to synchronize an identical chaotic system at the receiver end. For suitable values of the parameters, the receiver chaotic system variables synchronize with those of the transmitter. The synchronization also allows the regeneration of the chaotic key without actually transmitting it. Thus the information signal is recovered by employing the decryption rule (same modulo two-addition) on the encrypted signal with the regenerated chaotic

digital sequence. For example, a logistic map $X_{n+1} = \mu X_n(1-X_n)$ of the interval $0 \le X_n \le 1$ onto itself, with control parameter μ varying between $3.6 \le \mu \le 4$ with the exception of periodic windows, can be considered for chaotic encryption. After generation of the encrypted digital signal as described above, the signal is passed through the error correcting (convolutional) encoder, the rate of which is decided according to the application. The rate of the convolutional code used will directly affect the amount of payload associated with the message. Since the encrypted message signal is encoded using a convolutional encoder sequence and added with the encrypted message signal during the encoding process, a maximum-likelihood decoding operation is done at the receiver to produce an estimate of the encrypted message signal. By use of a sequential decoding algorithm to estimate the original encrypted message signal from the received signal, an estimate of the embedded signal is produced. If there is an extra stage of an interleaver at the transmitter, the output of the decoder is given to the deinterleaver which orders the data in the correct sequence. Finally the signal is decrypted by synchronizing with the chaotic generator which is used during encryption.

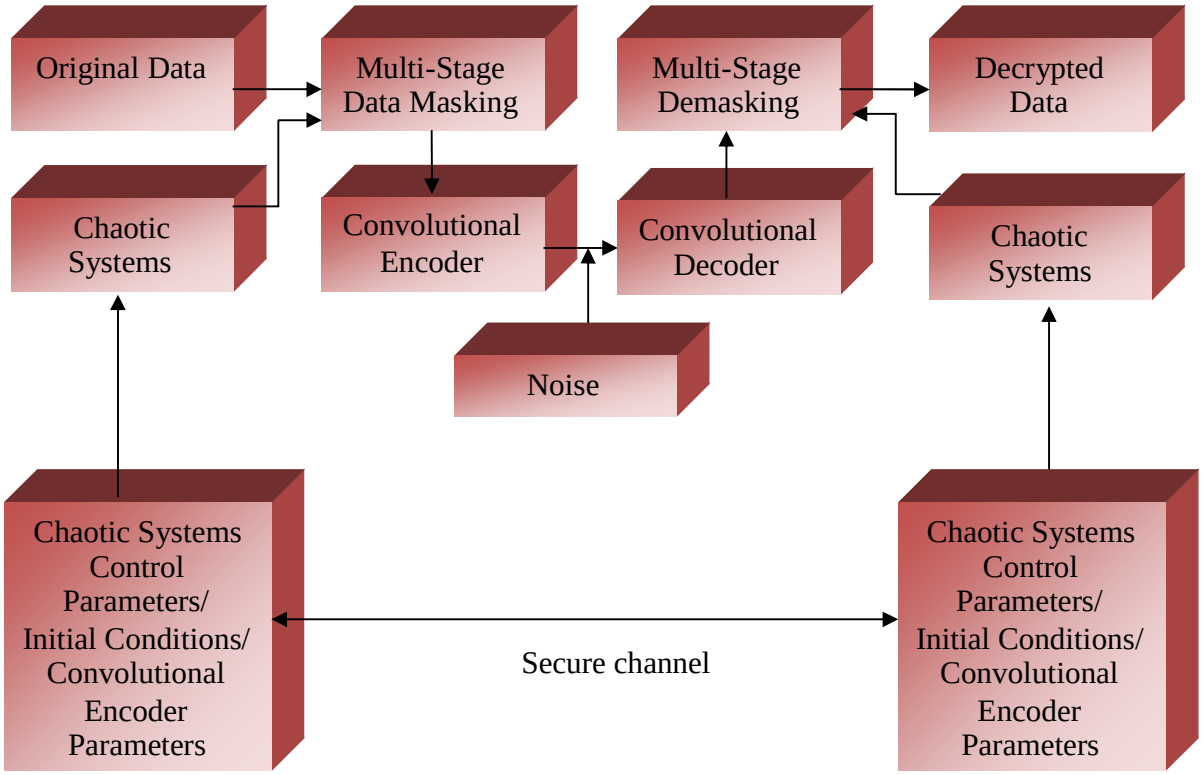


Figure 9.3 Synchronization in a chaotic cryptosystem with a convolutional coding.

9.4 Convolutional Coding Theory

A communication system channel is subject to various types of noise, distortion, and interference. The available amount of transmitter power and bandwidth are the major constraints in the design of a communication system. Therefore, we need some form of error control encoding to recover the information reliably. Error-control coding is used to achieve reliable communication in presence of interference and to protect information from intentional interference. Shannon's coding theorems in 1948 claim that any communication channel can be characterized by a maximum theoretical capacity. If the source information rate is less than the maximum theoretical capacity, then there exists a channel-coding method such that information can be reliably transmitted. This theorem set a theoretical limit on the possible information rate

for achieving error-free transmission through a channel by appropriate coding. Otherwise, reliable transmission is not possible regardless of amount of signal processing performed at the transmitter and receiver. He not only established the limits of the gains possible with coding, but also proved the existence of codes that could effectively reach these limits. Shannon showed that for every communication channel, there is a constant *C*, called the capacity of the channel, which has the following fundamental significance: if one wishes to communicate over the channel at a rate *R*, then one can do so as reliably as desired, if and only if $R < C$. Specifically, for every $\delta > 0$ there exists a sufficiently long error-correcting code of rate *R*, such that the probability of error in maximum-likelihood decoding is at most δ. Thus if the required transmission rate is less than *C*, it is possible to design a communication system for that channel and with the help of error-control coding one can achieve a very small probability of output error for that channel. The capacity *C* in bits per second depends on only two parameters, the channel bandwidth and the signal-to-noise ratio. In practice, it has proved to be remarkably difficult to find classes of constructive codes that can be decoded by feasible decoding algorithms at rates which come at all close to the Shannon limit. Within the past decade there have been remarkable breakthroughs, principally the invention of turbo codes and the rediscovery of Low-Density Parity Check codes, which have allowed the capacity of additive white Gaussian noise (AWGN) channels to approach, in a practical sense, close to the theoretical Shannon limit than any other code so far.

There are many different types of error control codes like BCH codes, Reed Solomon codes, Linear Block codes, Turbo codes and Convolutional codes. Different factors affect the choice of a particular coding scheme. Constraints like cost, power, bandwidth, and types of channel, allowable delay in decoding, data rate and type of information, play a major role in selection of a particular coding scheme. Convolutional error correcting codes are extensively

used for real time error correction in telecommunication systems. For example, these codes are used in communication systems like GSM (global system for mobile communications), in Interim Standard IS-95 on forward traffic channels with a rate 1/2 encoder, in 802.11/a HiperLAN-2 (High-performance Local Area Network type 2) with rate 1/2 for 9 Mbps BPSK (Binary Phase Shift Keying) modulation and rate 2/3 for 54 Mbits/s 64-QAM (64 level Quadrature Amplitude Modulation) transmission. It has been the predominant Forward Error Correcting (FEC) technique used in geostationary satellite communication networks, such as Very Small Aperture Terminal (VSAT) networks. In VSAT generally a rate-1/2 convolutional coding with constraint length 7 is used. With this code, one can transmit BPSK or QPSK (Quadrature Phase Shift Keying) modulated signals with at least 5 dB less power than is needed without it. The first large-scale applications include a rate-1/2 convolutional code with constraint length 20 for the Pioneer 1968 mission. The receiver used 3-bit soft decisions and sequential decoding implemented on a general-purpose 16-bit minicomputer with a 1 MHz clock rate. At 512 bits/s, the actual coding gain achieved at P_b=0.005, was about 3.3 dB. If the modulation technique stays the same, the bandwidth expansion factor of a convolutional code will be *n/k*. Convolutional codes differ from block codes in that each encoding operation depends on both current and a number of previous information groups. Convolutional code has some advantage in building self-orthogonal code and punctured code.

Another example is in the digital architecture of a mobile transmitter that supports Wideband Code Division Multiple Access (WCDMA). To conform to the WCDMA standard, Cyclic Redundancy Code (CRC) bits are added for error detection, and error correction bits are added for channel coding. The standard defines two encoding schemes to support a different quality of services. For voice and MPEG4 applications, the standard employs convolutional

encoding. For data applications, the standard uses turbo encoding. Turbo encoding yields a relatively large encoding gain with a reasonable computational complexity. This encoding scheme is useful for data services that permit longer transmission delays. The data are then interleaved. The coded symbols are written into the interleaver. Interleaving is a standard practice to combat signal degradation due to burst errors on the channel [118]. The data are then spread with a user or channel specific Orthogonal Variable Spreading Factor (OVSF) code to produce a data stream at a given chip-rate, where the cross correlation among all users is zero. The spread data stream is scrambled with Gold code so that the multipath signals can be uniquely identified and decoded by the receiver. To transmit a signal within the specified bandwidth, the data bits are shaped by using a pulse shaping filter. Next, the signal passes through carrier modulation and up-conversion to radio frequency. Each channelizer accesses the digital IF and translates a channel to the baseband. The radio environment of a wireless network system is multipath. To be effective, the system requires a despreader that can simultaneously despread the numerous multipaths of a single user, as well as multiple users (for a joint detection). A RAKE receiver, with its multiple fingers to despread the different multipaths, is employed for this function. Then, the block deinterleaver is responsible for performing the reverse action of the interleaver of the transmitter. There are two types of decoders that are used in the receiver. The Viterbi decoder is used to decode signals encoded by the convolutional encoder, and the turbo decoder is used with the turbo encoder. Figure 9.4 shows the general position of the channel encoder in the block diagram of a digital communication system. In next section we describe the encoder and Viterbi decoder structures for convolutional codes [119].

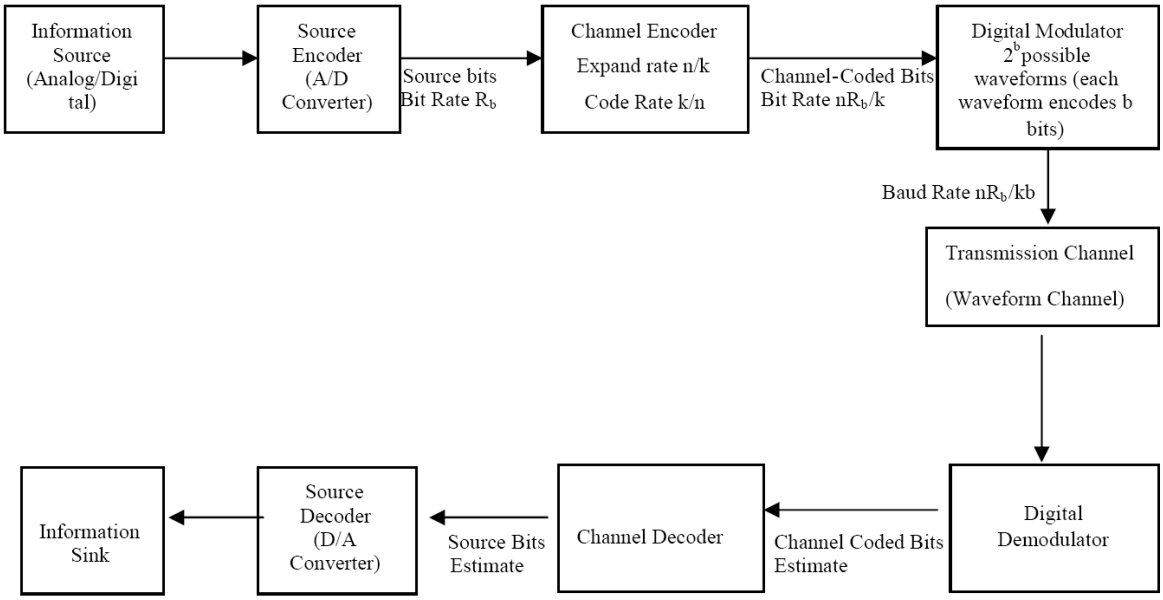


Figure 9.4 Channel encoder/decoder position in the block diagram of a digital communication system.

9.5 Convolutional Coding Theory and Implementation

Consider the information *k*-tuples at time *i* over a Galois field GF_q^k

$$\begin{aligned} \vec{x}_i &= (x_i^{(1)} \ldots x_i^{(k)}), \\ \vec{x}(D) &= \sum_{i=r}^{\infty} \vec{x}_i D^i . \end{aligned} \tag{9.1}$$

where $x(D)$ is the sequence of information *k*-tuples. Through the same process we can define the encoded information sequence as *n*-tuples at time *i* by

$$\begin{aligned} \vec{y}_i &= (y_i^{(1)} \ldots y_i^{(n)}), \\ \vec{y}(D) &= \sum_{i=r}^{\infty} \vec{y}_i D^i . \end{aligned} \tag{9.2}$$

In vector realization

$$\vec{x}(D) = \left(\vec{x}^{(1)}, \vec{x}^{(2)} \ldots \vec{x}^{(k)}\right),$$
$$\vec{y}(D) = \left(\vec{y}^{(1)}, \vec{y}^{(2)} \ldots \vec{y}^{(n)}\right). \tag{9.3}$$

Given a code C of length n and dimension k, one can encode an arbitrary information word $\vec{x} \in GF_q^k$ into a codeword $\vec{y} \in GF_q^n$ via a one-to-one mapping $\zeta : GF_q^k \to GF_q^n$ called an encoder for C. An encoder is a k by n convolutional encoder if the mapping $\zeta : GF_q^k \to GF_q^n$ realized by the encoder $G(D)$ and can be represented by $\vec{y}(D) = \vec{x}(D)G(D)$ where $G(D)$ is a $k \times n$ encoding matrix of rank k. This means an encoding scheme can be implemented in time $O(n^2)$ as a vector-matrix product.

$$G = \begin{pmatrix} G_0 & G_1 & \ldots & G_n & & \\ & G_0 & G_1 & \ldots & G_n & \\ & & G_0 & G_1 & \ldots & G_n \\ & & & \ldots & \ldots & \ldots \end{pmatrix}.$$

The primary example of a finite field in theory and in practice is $F_2 = \{0,1\}$, and codes over F_2 are called binary. A code is said to be linear if it is a subspace of GF_q^n. The rate of a linear code $C \subseteq GF_q^n$ is defined as k/n. Thus the encoder rate is the number of information symbols per code symbol. The design purpose is to make this rate to be as high as possible. The mechanism of convolutional encoder is a binary Galois field GF_q^n with a k -input, n-output finite-state machine where q is the number of bits in a group. Thus, the set of k data groups, each of a fixed length q, is input into an (n, k, K) convolutional encoder, and ($n - k$) redundant packets are generated based on a generator matrix. The parameter K refers to the memory of the encoder, and indicates how many previous code groups influence the redundant packet.

The error-correction capability of a code has to do with its minimum Hamming distance. Let $\vec{x}, \vec{y} \in GF_2^n$. The Hamming distance $d_H(x,y)$ between $\vec{x}$ and $\vec{y}$, given by

$$d_H(\vec{x}, \vec{y}) \equiv \underset{\substack{x,y \in C \\ x \neq y}}{w_H(\vec{x} - \vec{y})}, \tag{9.4}$$

is the number of positions where $\vec{x}$ and $\vec{y}$ differ. For instance $d_H(\vec{x},\vec{y}) = d_H(10101,01010) = 5$. The Hamming weight $w_H(\vec{x}) = d_H(\vec{x},0)$, is the number of nonzero positions in $\vec{x}$. $w_H(\vec{x}) = d_H(\vec{x},\vec{0}) = d_H(10101,00000) = 3$. The Hamming distance is a metric for the vector space GF_2^n since it satisfies the conditions for positive definiteness and symmetry, and the triangle inequality. The Hamming weight and Hamming distance definitions in GF_2^n can be generalized to *n*-tuples over GF_q^n. The maximum error-correcting capability t_{free} of a code or an encoder is determined by its free distance and given by following relation:

$$t_{free} = \left\lfloor \frac{(d_{free} - 1)}{2} \right\rfloor. \tag{9.5}$$

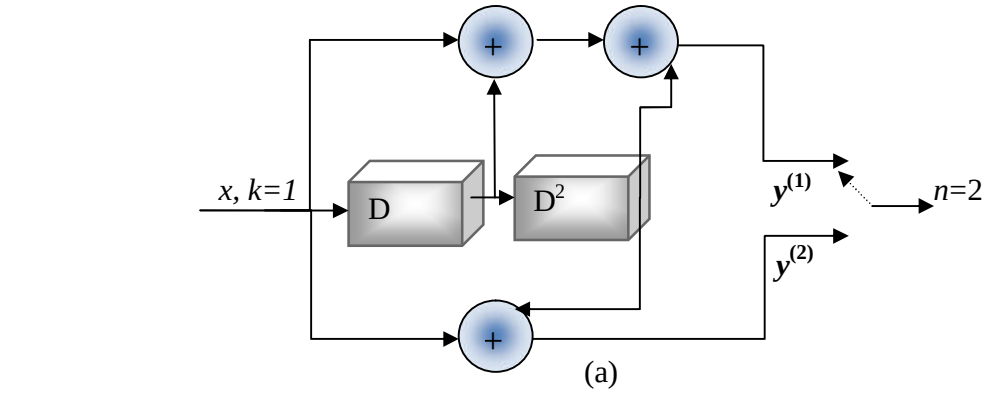
Thus an error correcting code can be viewed as a packing of disjoint spheres of radius t_{free} in the space GF_2^n endowed with the Hamming metric. If code words of *C* are transmitted over a noisy channel, then errors in any position $\leq t_{free}$ may be corrected at the receiver end by identifying the unique sphere to which the error-corrupted channel output belongs. Evidently, we would like *k/n*, and d_{free} to be large. Attaining a high rate and a large distance are conflicting goals. Hence, the codes of interest for communications are those that achieve a good tradeoff between these two parameters. The most important distance property of convolutional codes is the free distance. The free distance, d_{free}, of a convolutional code is the minimum Hamming

distance between two code sequences, $d_{free} \equiv \min_{\vec{x},\vec{y}\in C}\{d_H(\vec{x},\vec{y})\}$. Since convolutional codes are linear, all non-zero code sequences can be compared with the all-zero sequence to get the same result, $d_{free} \equiv \min_{\vec{y}\in C\setminus 0}\{w_H(\vec{y})\}$. The free distance determines the error correction capability of the convolutional code. The minimum free distance corresponds to the ability of the convolutional code to estimate the best decoded bit sequence. A minimum distance decoder can always correct an error sequence, $\vec{\varepsilon}$, if $w_H(\vec{\varepsilon}) < \frac{d_{free}}{2}$. As d_{free} increases, the performance of the convolutional code also increases. A variety of techniques used to search for and find good convolutional codes. The quality factor is always a measure of the weight structure of the code. When maximum likelihood decoding is used, the optimum weight structure is that which has the minimum number of bit errors in the paths through the code trellis which are closest to one another in Hamming distance. The weight structure of the best rate-1/*n* codes, where *n* = 2 , 3, 4, 5, 6, 7, in this sense were introduced and analyzed extensively in this chapter. The constraint length of the code is denoted by *K* and the number of states in the code trellis is 2^{K-1}. The free distance is the minimum Hamming distance between the code words on any two paths through the trellis.

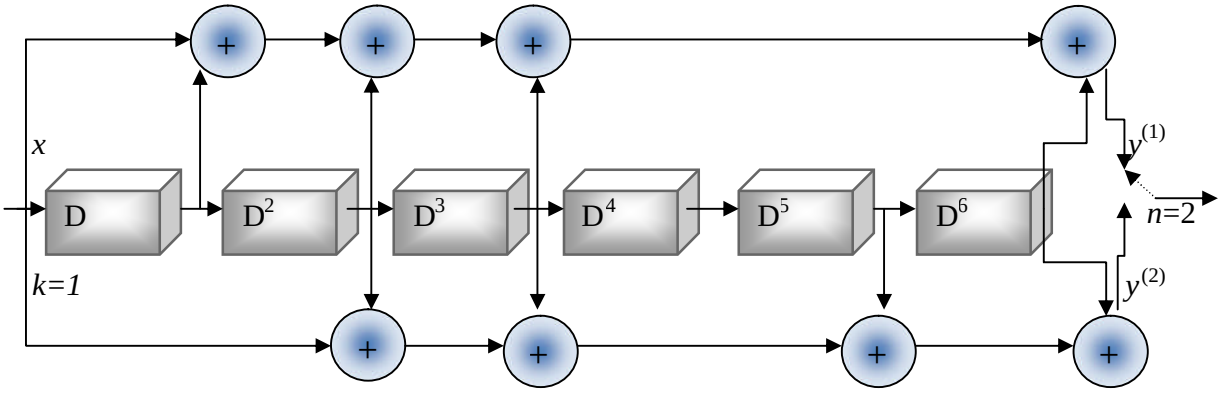
The encoder will be represented in many different but equivalent ways. There are a generator representation, tree diagram representation, a state diagram representation and a trellis diagram representation. A generator representation shows the hardware connection of the shift register taps to the modulo-2 adders. A generator vector represents the position of the taps for an output. A one represents a connection and a zero represents no connection. Convolutional encoder introduces redundant bits into the data stream through the use of linear shift registers. The input data to the encoder is shifted into and along the shift register, *k* bits at a time. *k* is the

number of parallel input information bits and *n* is the number of parallel output encoded bits at one time interval. Then the code rate is *k/n*. The constraint length *K* is the number of memories in the shift registers. The shift registers store the state information of the convolutional encoder and the constraint length relates the number of bits upon which the output depends. The code generators are given in octal notation. This notation gives the connection between the encoder shift register stages and the module-2 adders. We will use $C^{(k/n)}(K, [G_1(D), G_2(D), \ldots G_i(D)])$ notation to include all information in this code. For example, in encoder $C^{(1/2)}(2, [7, 5])$ or $G(D)=(G_1(D), G_2(D))=(1+D+D^2, 1+D^2)$, the code rate is $R = k/n = 1/2$ which means two bits are output for every single bit that is input and the constraint length is $K=2$ (Figure 9.5 (a)). For the encoder $C^{(1/2)}(7, [171, 133])$ in Figure 9.5 (b) the constraint length 7 means there are seven stages in the shift register. In binary the generators are presented as [1111001, 1011011]. This means that the connections to the first module-2 adder are from shift register stages 0, 1, 2, 3, and 6 to the second module-2 adder are from shift register stages 0, 2, 3, 5 and 6. In a polynomial representation $G(D)=[G_1(D), G_2(D)]$, terms like $G_1(D)$ and $G_2(D)$ represent a shorthand that defines the location of the taps on the shift register. As shown in the Figure 9.5 (b), $G_1(D)=171$ represents the octal code for the upper connections to the shift register while $G_2(D)=133$ describes the lower connections. $G_1(D)=(1+D+D^2+D^3+D^6)=(171)_8$,

$G_2(D)=(1+D^2+D^3+D^5+D^6)=(133)_8$.



(a)



(b)

Figure 9.5 (a) An encoder in controller form of the generator $C^{(1/2)}(2, [7, 5])$ or $G(D)=[G_1(D), G_2(D)]=[1+D+D^2, 1+D^2]= [(111)_2,(101)_2]= [(7)_8,(5)_8]$; (b) An encoder in controller form of the generator $C^{(1/2)}(7, [171, 133]$. In binary the generators are presented as [1111001, 1011011]. $G(D)=[G_1(D), G_2(D)]=[(1+D+D^2+D^3+D^6), (1+D^2+D^3+D^5+D^6)]=[(171)_8, (133)_8]$.

It is often helpful to view the information sequences and code sequences in a tree structure. In the tree diagram, a solid line represents input information bit 0 and a dashed line represents input information bit 1. The corresponding output encoded bits are shown on the branches of the tree. An input information sequence defines a specific path through the tree diagram from left to right. Each input information bit corresponds to branching either upward (for input information bit 0) or downward (for input information bit 1) at a tree node. If the input

is 0 choose the upper branch and if it is 1 choose the lower. The labels of the branches are the corresponding code symbols and the nodes are labeled with the encoder states. In Figure 9.6 such tree is shown for the generator matrix $G(D)=(1+D+D^2, 1+D^2)$. In the figure time passes from left to right. The nodes are labeled with the corresponding states of the encoder. The encoder is realized in controller canonical form and starts in the zero state. The state represents all the encoder knows about the past. Therefore, the continuation of two paths with the same state at same time will be identical.

The state diagram shows the state information of a convolutional encoder. The state information of a convolutional encoder is stored in the shift registers. In the state diagram, the state information of the encoder is shown in the circles. Each new input information bit causes a transition from one state to another. The path information between the states, denoted as *x/y*, represents input information bit *x* and output encoded bits *y*. It is customary to begin convolutional encoding from the all zero state. The trellis diagram is basically a redrawing of the state diagram. It shows all possible state transitions at each time step. Frequently, a legend accompanies the trellis diagram to show the state transitions and the corresponding input and output bit mappings (*x/y*).

9.6 Convolutional Code Performance

The performance of convolutional codes can be quantified through analytical means or by computer simulation. The analytical approach is based on the transfer function of the convolutional code which is obtained from the state diagram. The analysis of convolutional codes is generally difficult to perform because traditional algebraic and combinatorial techniques cannot be applied. These codes can be analyzed through their transfer functions. By utilizing the state diagram, the transfer function can be obtained. With the transfer function, code properties

such as distance properties and the error rate performance can be easily calculated. From the transfer function analytically the minimum free distance is determined as the lowest exponent of *D* in the encoder transfer function. The transfer function is obtained from the state diagram. The analysis of convolutional codes is generally difficult to perform because traditional algebraic and combinatorial techniques are not applicable. With the transfer function, code properties such as distance properties and the error rate performance can be calculated.

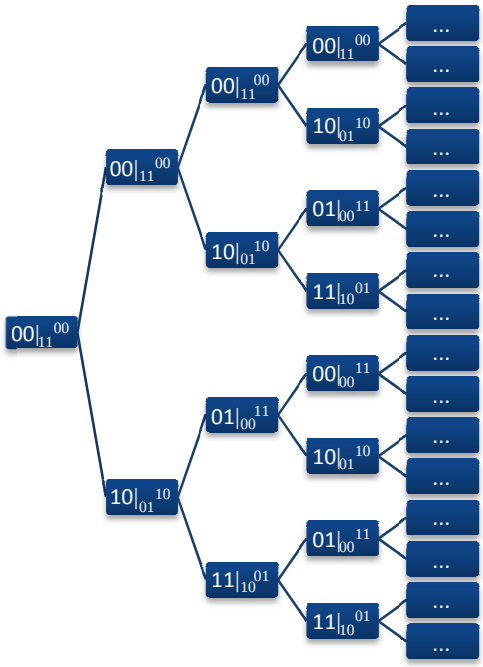


Figure 9.6 A tree structure representing the generator matrix $G(D)=(1+D+D^2, 1+D^2)$.

To obtain the transfer function, the following rules are applied: 1) Break the all-zero (initial) state of the state diagram into a start state and an end state. This will be called the modified state diagram. 2) For every branch of the modified state diagram, assign the symbol *D* with its exponent equal to the Hamming weight of the output bits. 3) For every branch of the modified state diagram, assign the symbol *J*. 4) Assign the symbol *N* to the branch of the

modified state diagram, if the branch transition is caused by an input bit 1. For a convolutional code described by, g_1=[1,1,0], g_2=[1,0,1], g_3=[1,1,1] in Figure 9.7 (a), the state transition diagram is depicted in Figure 9.7 (b).

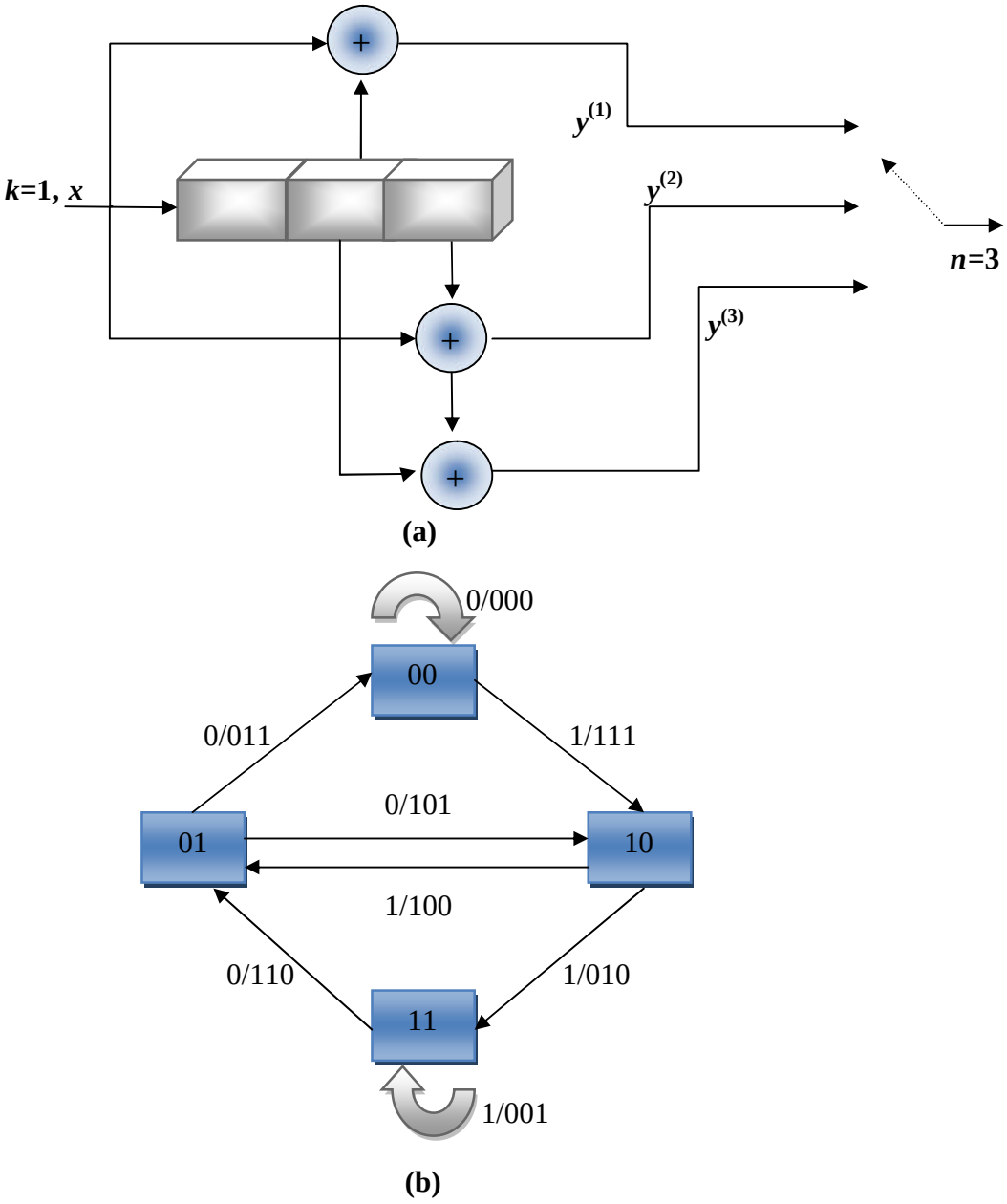


Figure 9.7 Convolutional code g_1=[1,1,0], g_2=[1,0,1], g_3=[1,1,1] with k=1, n=3, K=3 (a) encoder (b) state diagram.

The trellis diagram for this code for two frames of the trellis associated with the code is shown in Figure 9.8 (a). Also the diagram used to find the transfer function is shown in Figure 9.8 (b). Solid lines indicate an input equal to zero, whereas dotted lines correspond to an input equal to one.

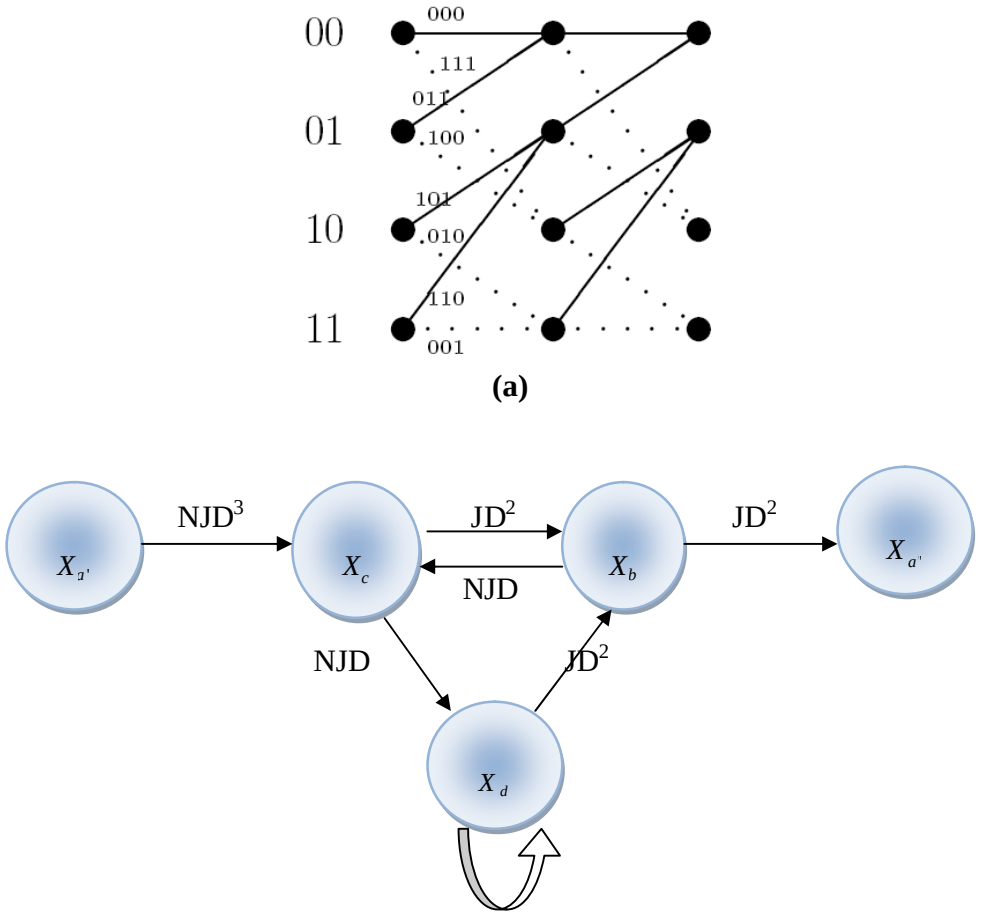


Figure 9.8 Convolutional code g_1=[1,1,0], g_2=[1,0,1], g_3=[1,1,1] with k=1, n=3, K=3 (a) Trellis diagram (b) State diagram.

Using the flow graph results, the system transfer function can be written as:

$$\begin{aligned} X_c &= D^3NJX_{a'} + DNJX_b, \\ X_b &= D^2JX_c + D^2JX_d, \\ X_d &= DNJX_c + DNJX_d, \\ X_{a''} &= D^2JX_b. \end{aligned} \tag{9.6}$$

Eliminating X_b, X_c, and X_d results in:

$$T(N,D,J) = \frac{X_{a''}}{X_{a'}} = \frac{D^7NJ^3}{1-DNJ-D^3NJ^2}. \tag{9.7}$$

The free distance of the code is determined by setting $N=J=1$ in the transfer function, so that

$$\begin{aligned} &T_1(D) = T(D,N,J)\big|_{N=J=1} = \frac{D^7}{1-D-D^3} = D^7 + D^8 + D^9 + \cdots, \\ &d_{free} = 7. \end{aligned} \tag{9.8}$$

Since the lowest order of D in polynomial $T_1(D)$ is 7, for this system d_{free} is 7. Also, when N and J are set to 1, the coefficients of D represent the number of paths through the trellis with weight D. More information about the codeword is obtained from observing the exponents of N and J.

For a codeword, the exponent of N indicates the number of 1s in the input sequence, and the exponent of J indicates the length of the path that merges with the all-zero path for the first time. Since there is no self loop corresponding to an input equal to one such that the output is the all zero sequence, the code is not a catastrophic convolutional code. Catastrophic convolutional code causes a large number of bit errors when only a small number of channel bit errors are received. This type of code needs to be avoided and as we mentioned can be identified by the state diagram.

9.7 Decoding Complexity

When a codeword $\vec{c}$ of a linear code $C \subset F_q^n$ is transmitted over a noisy communication channel with output alphabet A, and a vector $\vec{r} \in A$ is observed at the channel output, the decoding task will be to determine what $\vec{c}$ was if $\vec{r}$, C and a probabilistic model for the channel are known. There is a large number of various approaches to this task, which can be roughly classified into four categories, according to whether they are maximum-likelihood or bounded distance, and hard-decision or soft-decision. Hard and soft decision schemes are related to the channel model assumed for decoding purposes. In both cases, the channel is usually assumed to be memoryless. This means that the noise is an i.i.d. (independent, identically distributed) random process. That means the probability of error is the same at all times (codeword positions), and what happens at one time is independent of what happens at all other times. Hard-decision and soft-decision decoding refer to the type of quantization used on the received bits. Hard-decision decoding uses 1-bit quantization on the received channel values. Soft-decision decoding uses multi-bit quantization on the received channel values. For the ideal soft-decision decoding (infinite-bit quantization), the received channel values are directly used in the channel decoder. A convolutional encoder is basically a finite-state machine. Hence the optimum decoder is a maximum-likelihood (ML) sequence estimator. For a given code, maximum likelihood soft-decision decoding requires at least two dB less energy per bit than maximum-likelihood hard-decision decoding, to achieve the same probability of decoding error. However, the decoding of a code by a soft-decision scheme is much harder to perform than hard-decision decoder. Depending on whether the detector following the demodulator performs hard or soft decisions, the corresponding metric in the trellis search may be either a Hamming metric

or a Euclidean metric, respectively. Theoretically for a given channel output $\vec{y} \in A$, the optimal decoding strategy is to find the codeword $\vec{x} \in C$ that maximizes the probability $P(\vec{r}|\vec{y})$ that $\vec{c}$ was transmitted given that $\vec{r}$ was received. We may usually assume, without loss of generality, that the codewords of $\vec{c} \in C$ are transmitted with equal a priori probability. In this case, by a simple application of Bayes rule, the optimal decoding strategy is equivalent to finding the most likely codeword $\vec{y} \in C$ that maximizes the probability $P(\vec{r}|\vec{y})$ that $\vec{y}$ would be received if $\vec{r}$ was transmitted. A decoder for C that always finds the most likely codeword is said to be a maximum-likelihood decoder.

Hard-decision maximum-likelihood decoding is a nearest neighbor search in the space F_q^n endowed with the Hamming metric. The Viterbi algorithm selects $\vec{y}$ to maximize $P(\vec{r}|\vec{y})$. The channel is assumed to be memoryless, and thus the noise process affecting a received bit is independent from the noise process affecting all of the other received bits. The probability of joint, independent events is equivalent to the product of the probabilities of the individual events. Thus,

$$P(\vec{r}|\vec{y}) = \prod_{i=1}^{L+m-1} p\left(r_i^{(1)}|y_i^{(1)}\right) p\left(r_i^{(2)}|y_i^{(2)}\right) \ldots p\left(r_i^{(n)}|y_i^{(n)}\right).$$

This equation is the likelihood function of $\vec{y}$ given that $\vec{r}$ is received. The estimate that maximizes $P(\vec{r}|\vec{y})$ also maximizes $\log\{P(\vec{r}|\vec{y})\}$ because logarithms are monotonically increasing functions. Thus, a log likelihood function can be defined as:

$$\log\{P(\vec{r}|\vec{y})\} = \sum_{i=1}^{L+m-1} \left\{ \sum_{j=1}^{n} p\left(r_i^{(j)}|y_i^{(j)}\right) \right\},$$

For an easier manipulation of the summations over the log function, a bit metric is defined as follows

$$\mathrm{M}\left(r_i^{(j)}\middle|y_i^{(j)}\right)= a\left\{p\left(r_i^{(i)}\middle|y_i^{(j)}\right)+b\right\},$$

where a and b are chosen such that the bit metric is a small positive integer. Notice that maximum-likelihood decoding is a computational task that means different things for different channels. On a binary symmetric channel, the most likely codeword is obviously the one closest to $\vec{y}\in F_2^n$ in the Hamming metric. In fact, the same conclusion easily extends to the more general case of q-ary symmetric channels. For example, the values a and b for binary symmetric channel hard-decision decoding are defined based on the fact that $p\left(X=1|Y=0\right)= p\left(X=1|Y=0\right)=\varepsilon$ is the probability of transmitting bit 0 (1) and receiving bit 1 (0), while $p\left(X=0|Y=0\right)= p\left(X=1|Y=1\right)=1-\varepsilon$ is the probability of transmitting bit 0 (1) and receiving bit 0 (1), then a and b can be chosen as

$$a=\frac{1}{\log\varepsilon-\log(1-\varepsilon)},$$
$$b=-\log(1-\varepsilon).$$

The bit metric is written as,

$$\mathrm{M}\left(r_i^{(j)}\middle|y_i^{(j)}\right)=\frac{1}{\log\varepsilon-\log(1-\varepsilon)}\left\{p\left(r_i^{(i)}\middle|y_i^{(j)}\right)-\log(1-\varepsilon)\right\}.$$

This bit metric shows the cost of receiving and decoding bits and is known as the Hamming distance metric. Thus, the Viterbi algorithm chooses the code $\vec{y}$ that has the largest Hamming distance relative to received sequence $\vec{r}$. Now we can define the path metric as the total cost of estimating the received bit sequence $\vec{r}$ with the decoded bit sequence $\vec{y}$ in the trellis diagram.

$$\mathrm{M}\left(\vec{r}\middle|\vec{y}\right)=\sum_{i=0}^{L+m-1}\sum_{j=1}^{n}\mathrm{M}\left(r_i^{(j)}\middle|y_i^{(j)}\right).$$

The k^{th} branch metric presents the cost of choosing a branch from the trellis diagram and the k^{th} partial branch metric shows the cost of choosing a partially decoded sequence $\vec{y}$ up to time *k*. These are given by

$$\mathrm{M}\left(\vec{r}_k \left| \vec{y}_k \right.\right) = \sum_{j=1}^{n} \mathrm{M}\left(r_k^{(j)} \left| y_k^{(j)} \right.\right),$$

$$\mathrm{M}^{(k)}\left(\vec{r} \left| \vec{y} \right.\right) = \sum_{i=0}^{k} \mathrm{M}\left(\vec{r}_i \left| \vec{y}_i \right.\right).$$

The Viterbi algorithm utilizes the trellis diagram to compute the path metrics. A partial path metric is assigned to each state (node) in the trellis diagram. The partial path metric is determined from state S_0 at time 0 to a particular state S_k at time. At each state, the best partial path metric is chosen from the paths terminated at that state. The best partial path metric may be either the larger or smaller metric, depending whether *a* and *b* are chosen conventionally or alternatively. The selected metric represents the survivor path and the remaining metrics represent the non-survivor paths. The survivor paths are stored while the non-survivor paths are discarded in the trellis diagram. The Viterbi algorithm selects the single survivor path left at the end of the process as the ML path. Trace-back of the ML path on the trellis diagram would then provide the ML decoded sequence.

The soft-decision maximum-likelihood decoding (for binary codes) is equivalent to finding the codeword $\vec{y} \in C$ which maximizes the log-likelihood sum metric $PM^{(i)} = \sum_{j=1}^{B} \mu_j^{(i)}$, where $\mu_j^{(i)} = \log\left\{P\left(\vec{r} \left| \vec{y} \right.\right)\right\} = \sum_{i=1}^{L+m-1} \left\{ \sum_{j=1}^{n} p\left(r_i^{(i)} \left| y_i^{(j)} \right.\right) \right\}$, is the log-likelihood ratio for the j^{th} position. The criterion for deciding between two paths through the trellis is to select the one having the larger metric. Thus the discarded path segments are the segments with the smaller of the two

metrics. This rule maximizes the probability of a correct decision or, equivalently, it minimizes the probability of error for the sequence of information bits. In the important special case of AWGN channels, maximizing $PM^{(i)}$ again reduces to the nearest neighbor search, except that now C is viewed as a subset of [1, -1]^n and the search is in the space $\Re^N$ endowed with the Euclidean metric. Thus in most cases, maximum-likelihood decoding is equivalent to nearest neighbor decoding. Although maximum-likelihood decoding is the optimal decoding strategy, it is NP-hard for the general class of linear codes. Moreover, polynomial-time maximum-likelihood decoding algorithms are not known today for any specific family of useful codes, such as the binary BCH codes. Maximum likelihood decoding of a binary linear (n, k) code C, both hard-decision and soft-decision, can be trivially accomplished in time O(2^k) by simply comparing the channel output y with all the 2^k code words of C. Finally, it is noteworthy to mention that there are decoding strategies that are sub-optimal with efficient decoding algorithms. The major example of such a decoding strategy is bounded-distance decoding. A decoder is said to be bounded-distance if there exists a constant $t > 0$ such that the decoder always finds the closest codeword to a channel output $\vec{r}$, provided the distance from $\vec{r}$ to that codeword is at most t. There is no guarantee on what a bounded-distance decoding algorithm does if the distance from $\vec{r}$ to the closest codeword $\vec{y}$ exceeds t, it may still find $\vec{y}$, or it may output another codeword, or it may simply halt indicating failure [120, 121].

9.8 Decoding Complexity for Convolutional Codes

The input information sequence for a general convolutional code contains $k(K\text{-}1)$ bits. k is the number of parallel information bits at one time interval. K is the number of time intervals. This results in $2^{k(K-1)}$ states in the trellis diagram and exactly $2^{k(K-1)}$ distinct paths in the trellis diagram. Consequently, a search for the maximum likelihood sequence would have a

computational complexity on the order of $[2^{k(K-1)}]$. Viterbi algorithm is the decoding method for convolutional codes at the receiver end. The Viterbi algorithm reduces this complexity by performing the maximum likelihood search one stage at a time in the trellis. There are 2^k calculations at each node of the trellis. Therefore the number of nodes per stage in the trellis is $2^{(K-1)}$. As a result the complexity of the Viterbi algorithm is on the order of $[(2^k)(2^{K-1})]$. This is the number of calculations required to implement the maximum-likelihood decoder. Increasing either *k* or *K* will exponentially increase the decoding computational complexity [122].

9.9 Convolutional Coding Viterbi Decoding Algorithm

The steps involved in simulating a communication channel using convolutional encoding and Viterbi decoding in MATLAB are as follows. The data to be transmitted through the channel are generated using a character to string converter. We have generated 16, 32, 64, 128, …, 2046 bits from a given message. In the case of an image the binary value for the image is used. Convolutionally encoding the data is done in two steps. In the first step, the *poly2trellis* function is used. It accepts a polynomial description of a convolutional encoder and returns the corresponding trellis structure description. A polynomial description of a convolutional encoder describes the connections among shift registers and modulo-2 adders. In the next step the output of *poly2trellis* is suitable as an input to *convene* and *vitdec* functions. The parameters for the *poly2trellis* function are the constraint length and the code generator polynomial. The output of this is then used as one of the parameters for the *convene* function, along with the original data sequence. The convene function actually encodes the data bits. The *AWGN* function adds white Gaussian noise to the channel symbols produced by the encoder. The parameters for this function are the *coded symbols, SNR* (signal-to-noise ratio), the state, and the power type (whether in *dB* or *linear*). Two other factors also determine the performance of the Viterbi decoder. They are

commonly referred to as the decoding depth and the degree of quantization of the received signal. The decoding depth is a window in time that makes a decision on the bits at the beginning of the window and accepts bits at the end of the window for metric computations. This scheme gives up the optimum maximum likelihood decoding for the gain of using less memory and having a smaller decoding delay. It has been experimentally found that if the decoding depth is 5 times greater than the constraint length *K* then the error introduced by the decoding depth is negligible. The decoding is done for two different cases. The first one is hard decision decoding and the second is soft decision decoding. The *vitdec* function is used for this purpose. The input parameters required for this function are *code, trellis, tblen* (trace back length), *opmode* (operation mode) and *dectype* (decoder type). For *opmode* we have used a truncated mode of operation option *trunc.* In this the encoder is assumed to have started at the all-zero state. The decoder traces back from the state with the best metric. The *dectype* can have three alternatives, *unquant, hard* and *soft*. For hard decision decoding we use *hard, soft* for soft decision and *unquant* for unquantized decoding. Besides selecting *soft* for *dectype*, parameter *nsdec* need to be defined since our code consists of integers between 0 and $2^{(nsdec-1)}$. Before performing the decoding step, this can be determined by the number of quantized levels for received channel symbols. For hard decision decoding, the symbols are quantized to one bit precision while for soft decision decoding, data bits are quantized to three or four bits of precision. In our simulation we have used three bits (i.e. eight levels). The selection of quantization levels is an important design decision because of its significant effect on the performance of the link. For soft-decision Viterbi decoding, the degree of the quantization on the received signal can affect the decoder performance. The performance of the Viterbi decoder improves with higher bit quantization. It has been found that an eight-level quantizer degrades the performance only slightly with respect

to the infinite bit quantized case.

There are two error probabilities associated with convolutional codes, namely first event and bit error probabilities. The first event error probability, P_e, is the probability that an error begins at a particular time. The bit error probability, P_b, is the average number of bit errors in the decoded sequence. Usually, these error probabilities are defined using Chernoff Bounds. For hard-decision decoding, the first event error and bit error probabilities are bounded as in (Eq. 9.9) and (Eq. 9.10).

$$P_e < \sum_{d=d_{free}}^{\infty} a_d \sqrt[d/2]{4p(1-p)} < T(D)\Big|_{D=\sqrt{4p(1-p)}}. \tag{9.9}$$

The coefficients $\{a_d\}$ are the number of paths corresponding to the set of distances $\{d\}$. The upper bound on bit error probability is given by:

$$P_b < \left(\frac{1}{k}\right)\frac{dT(N,D)}{dN}\Big|_{N=1,D=\sqrt{4p(1-p)}}. \tag{9.10}$$

For soft decision decoding, the first event error and bit error probabilities can be expressed as:

$$P_e < T(D)\Big|_{D=e^{-\gamma_b R_c}},$$

where γ_b is the received SNR per bit and R_c is the code rate. A more useful measure of performance is the equivalent bit error probability. For the code bits transmitted by binary coherent PSK or four-phase coherent PSK, the bit error probability is expressed as:

$$P_b < \sum_{d=d_{free}}^{\infty} \frac{1}{k}\beta_d P_2(d) < \sum_{d=d_{free}}^{\infty} \frac{1}{k}\beta_d D^d\Big|_{D=e^{-\gamma_b R_c}} < \left(\frac{1}{k}\right)\frac{dT(N,D)}{dN}\Big|_{N=1,D=e^{-\gamma_b R_c}}, \tag{9.11}$$

where $P_2(d)$ is the probability of error in the pair wise comparison of two paths that differ in d bits. Assuming that a message has been encoded by the convolutional encoder of Figure 9.7 and

transmitted over a binary-symmetric channel with error probability of p=10^{-5}, we estimate an upper bound to the bit error probability. The transfer function is given by:

$$T(N,D,J)=\frac{X_{a''}}{X_{a'}}=\frac{D^7NJ^3}{1-DNJ-D^3NJ^2},$$
$$T(N,D,J=1)=\frac{D^7N}{1-DN-D^3N}. \tag{9.12}$$

Thus the bit error probability can be written as:

$$P_b \le \frac{dT(N,D,J=1)}{dN}\Big|_{N=1,D=\sqrt{4p(1-p)}},$$
$$\frac{d}{dN}\left(\frac{D^6N}{1-2D^2N}\right)=\frac{D^6-2D^8(1-N)}{(1-2D^2N)^2}, \tag{9.13}$$

$$P_b \le \frac{dT(N,D,J=1)}{dN}\Big|_{N=1,D=\sqrt{4p(1-p)}}=\frac{d}{dN}\left(\frac{D^6N}{1-2D^2N}\right)\Big|_{N=1,D=\sqrt{4p(1-p)}}$$
$$=\frac{D^6-2D^8(1-N)}{(1-2D^2N)^2}\Big|_{N=1,D=\sqrt{4p(1-p)}}=\frac{D^6}{(1-2D^2)^2}\Big|_{D=\sqrt{4p(1-p)}}=6.14\times10^{-14}.$$

This error is quite low. In simulation the *biterr* function is used for determining the number of errors and bit error rate. The input parameters for this function are the *original data sequence* and *the decoded sequence.* The number of errors can be easily obtained by simply subtracting these bit by bit. The ratio of the number of bits in error to the total number of bits gives us the bit error rate. Using the aforementioned steps for a given encoder, several simulations are performed to determine the effect of noise at different SNR values (ranging from 1 to 100). The results are plotted and compared using the *semilogy* function, which has a logarithmic scale for the vertical axis. Simulation has been performed again for different codes to determine if the decoding algorithm significantly affects a decoded sequence. The combined plot is useful to compare all the codes. Hard and soft decision decoding is compared for various codes. Here we have not included the steps of modulating the channel symbols onto a carrier and that of demodulating the

received carrier to recover the channel symbols. This is because even if we omit these steps, still we can accurately model the effects of the AWGN channel.

9.10 Simulation Results for Various Convolutional Codes

In this section, we investigate convolutional encoding and the Viterbi decoding to determine the bit error rate of the transmission channel without any particular modulation method. We compare its performance under different conditions. For both the Viterbi hard decision decoding method and soft decision decoding method, results suggest that soft decision decoding has at least 3 dB improvements in SNR compared to hard decision decoding for the same bit error rate. To exemplify and illuminate our approach, we have selected a communication channel with various rates. Also by using soft decision Viterbi decoding on an AWGN channel, we have examined various convolutional codes for different constraint lengths and rates.

Error performance analysis is performed by plotting the bit error-rate versus signal to noise ratio for an AWGN channel. Figure 9.9 show the simulation results. Encoders include rate-1/3 code with constraint length 3 and generator polynomial [4, 5, 7]; rate-1/2 code with constraint length = 7 and generator polynomial [171, 133]; rate-2/3 code with constraint length [4, 3] and generator polynomial [4 5 17; 7 4 2]; and finally, rate-2/3 code with constraint length [5, 4] and generator polynomial [23 35 0; 0 5 13]. For the codes used, our simulation result shows that, the soft decision decoding has at least 3 dB better responses compared to hard decision decoding. We find that the first code with rate 1/3 code and constraint length 3 has a better bit error rate performance with respect to other coding methods. This means, it gives a lower bit error rate for the same value of signal-to-noise ratio used in the other coding schemes. Thus, to achieve the same bit error rate the first code will require a lower SNR. A lower SNR

means a lower transmitter power. However, if we use this code, the bandwidth requirement is three times larger compared with an un-coded transmission. For the second coding scheme we have an increase in bandwidth by a factor of 2 and for the third and fourth coding scheme the increase factor is 1.5.

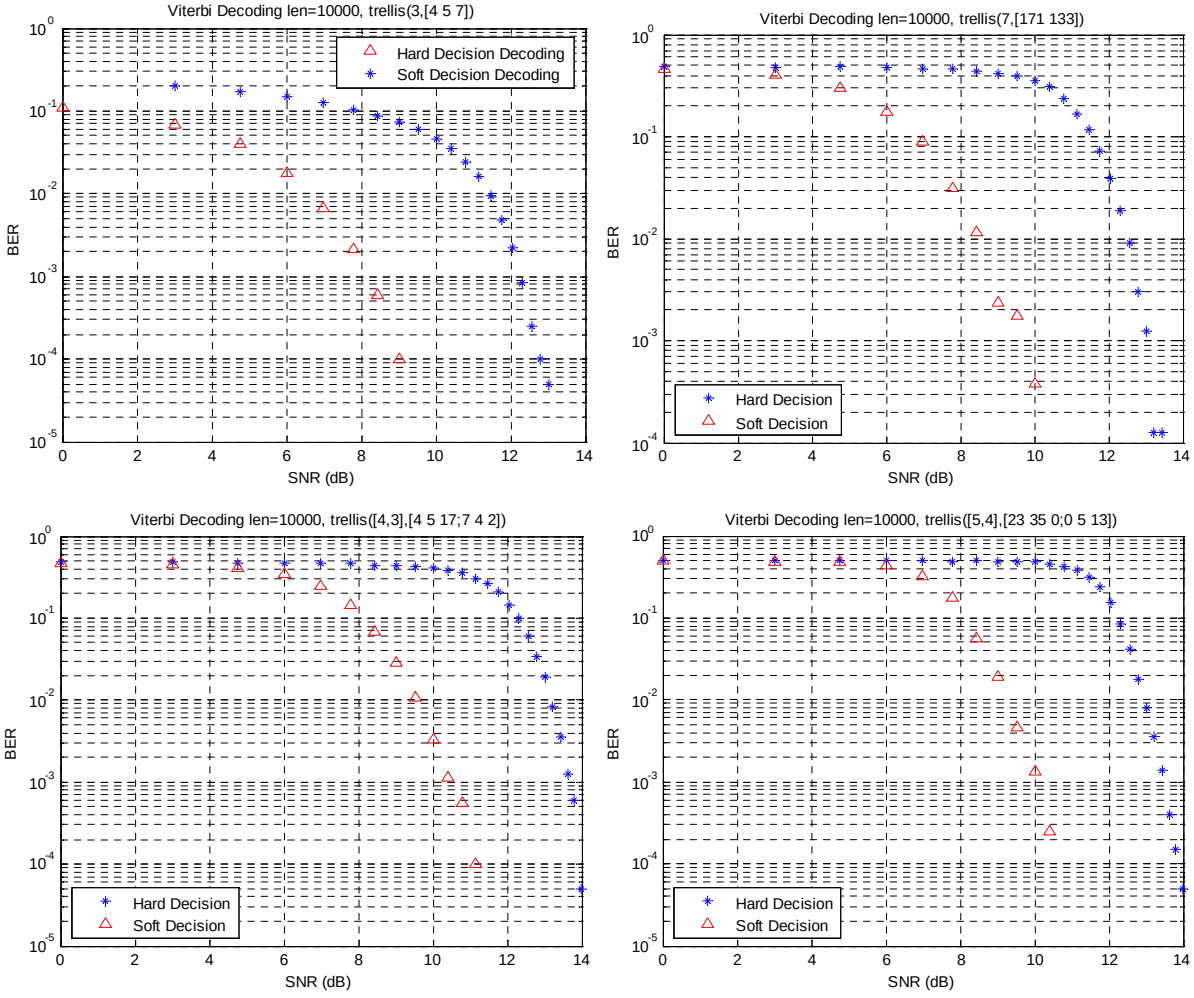


Figure 9.9 Simulation results for various convolutional codes with Viterbi (hard decision and soft decision) decoding on AWGN channel. (Top left) Rate 1/3 code with constraint length 3 and generator polynomial [4, 5, 7]. (Top right) Rate 1/2 code with constraint length=7 and generator polynomial [171, 133]. (Bottom left) Rate 2/3 code with constraint length [4, 3] and generator polynomial [4 5 17; 7 4 2]. (Bottom right) Rate 2/3 code with constraint length [5, 4] and generator Polynomial [23 35 0; 0 5 13].

Figure 9.10 shows the simulation result for various codes using only the soft decision Viterbi decoder. The five codes compared are:

(1) Rate 1/3 code with constraint length=3 and generator polynomial = [4, 3, 7].

(2) Rate 1/2 code with constraint length=7 and generator polynomial = [171, 133].

(3) Rate 1/2 code with constraint length=5 and generator polynomial = [37 33].

(4) Rate 2/3 code with constraint length= [4, 3] and

generator polynomial =[4 5 17; 7 4 2].

(5) Rate 2/3 code with constraint length=[5, 4] and

generator polynomial =[23 35 0; 0 5 13].

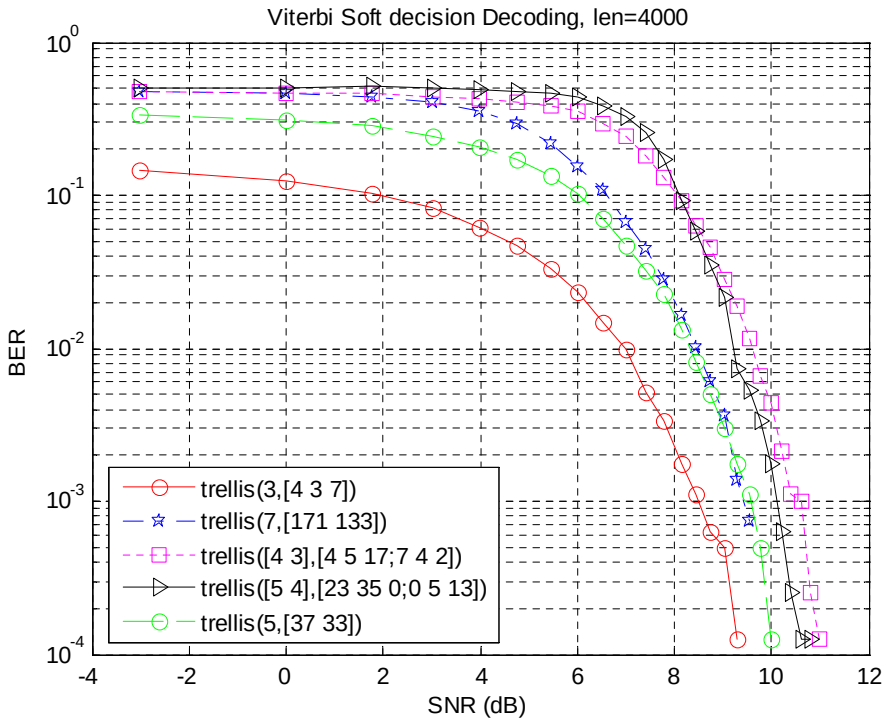


Figure 9.10 Simulation results for various convolutional codes (different constrain lengths and rate) using soft decision Viterbi Decoding on AWGN channel.

We find that the first code performance is better than the other three codes. It gives a lower BER for the same value of SNR. Thus to achieve the same BER, the first code will require

a lower signal-to-noise ratio, that is, lower transmitter power, compared to the other three codes. However if we use this code the bandwidth requirement is three times more compared with uncoded transmission (as the rate is 1/3). For the second and third code we have an increase in bandwidth by a factor 2 and for the fourth and fifth it is 1.5.

Convolutional coding is an effective method for trading bandwidth and implementation complexity against transmitter power. Convolutional codes are highly suitable for AWGN channels, where soft decision decoding is relatively straightforward. However, many types of conditions give rise to non-Gaussian conditions where soft decision decoding may need to adapt to the channel conditions and where the channel coherence may mean that Viterbi decoding is no longer the maximum likelihood solution. Using the aforementioned model and system architecture in section 9.3, several simulations are performed to compare the effect of convolutional encoder design, constraint length and encoder rate. In particular, using the Logistic system, which has been a paradigm in the study of chaotic systems from chapter 2, and 4-dim feedback-controlled hyperchaotic system studied in chapter 6, different best rate encoder architectures are used in this cryptosystem design. The results are compared to determine the significance of various design characteristics and in particular the message length. The simulator is configured to emulate each design for specific parameter values.

9. 11 Complexity of Chaotic Cryptosystem Convolutional Coding

Lempel-Ziv complexity is an important measure used in cryptography. Here we use this measure to test the randomness of the output of a chaotically masked convolutionally encoded message. This is done by using a symmetric cipher. We expect a random sequence of length $L(S_{random})$ has a close Lempel-Ziv complexity to the expected value of Lempel-Ziv complexity of a sequence of length L(S). However, the expected value of Lempel-Ziv complexity for arbitrary

length is unknown. We will use this Lempel-Ziv complexity measure in channel encoding methods with a chaotic masking cascade to measure the performance of each encoding algorithm with two specific chaotic maskings using a logistic map and a 4-dim feedback controlled system. This is because the dynamical complexity of the chaotic saddle of these systems is structurally stable with respect to variations in the noise-resisting strength. The main practical implication is that chaotic saddles embedded in a chaotic attractor can be naturally utilized as a noise-resisting but rich information source for a secure digital communication channel. Therefore the chaotic systems described by both one-dimensional chaotic map and four-dimensional hyperchaotic systems are used in simulation for chaotic encryption.

9. 11.1 Encryption with a Logistic Map

The Logistic system has been a paradigm in the study of chaotic systems. To facilitate a systematic numerical computation and analyses of the algorithmic complexity of the chaotic saddles, we make use of the logistic map $X_{n+1} = \mu X_n(1-X_n)$ which captures the essential dynamics of the one-hump maps arising in physical situations such as the Lorenz system. Consider the case where the map exhibits a chaotic attractor. The partition point for good symbolic dynamics is the critical point X_C = 0.5, where $f(X_n)$ is maximum. Trajectory points with $X_n > X_C, (X_n < X_C)$ bear the symbol 0 (1). That is, we assign a symbol 0 (1) to the trajectory if $X_n > 0.5$, $(X_n < 0.5)$. This critical point is often chosen to be the generating partition. The trajectory of points are generated on the chaotic saddle centered at the critical point X_C at μ=4 for which the logistic map apparently exhibits a chaotic attractor. This noise-resisting chaotic saddle is one of the uncountable infinite numbers of non-attracting chaotic sets that are embedded in the chaotic attractor. In principle, a suitable coding scheme in the symbolic dynamics of the logistic map is produced from this chaotic saddle.

In general, a chaotic saddle generated by a code is advantageous for communication, because the symbolic dynamics of the chaotic saddle is immune to small noise. Say, for example, that the system is in a noisy environment. If the original chaotic attractor is used to encode messages, a bit error that is 0 becomes 1 or vice versa can occur when the trajectory comes close to the partition point X_C because noise can kick the trajectory through X_C in both directions. However, trajectories that live on one of the infinite number of noise-resisting chaotic saddles do not come close to the partition point. The possibility for bit error due to noise can be substantially reduced when a chaotic saddle is utilized to encode messages, because there is a noise-resisting gap around the partition point X_C. Next, we give some illustrative examples of encoding an arbitrary message in the restricted chaotic signals. Suppose we wish to encrypt and encode, by using the Logistic attractor, the message ‘‘Beat Army!’’ in ASCII format. First, the original message is masked with the pseudo code generated by chaotic map. Thus, the encrypted message becomes MESSAGE XOR CHAOS. Then the output string is convolutionally encoded. We examined the original message containing the block of 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096 bits with seven bits for ASCII codes and the rest stuffed with zeroes in a row. The receiver can recover the original message simply by stripping the convoluted signal and demasking the output of the convolutional decoder. Many similar chaotic behaviors are observed for other parameter values of the logistic map, such as the one with well developed chaos after removal of transient responses initiated from a $X(1) = 0.345$ seed state at μ=4, on a 1/3-rate (3, [4 3 7]) encoder. Example trials using the ten different length sequences are shown in Figure 9.11. Here the effect of bit length in convolutional encoding and soft decision and hard decision Viterbi decoder bit error rate performance versus SNR (dB), with logistic chaos encryption, can

be identified. We also extend this idea to hyperchaotic 4-dim feedback-controlled system since it can be physically realized by an electronic Op-Amp circuit [123].

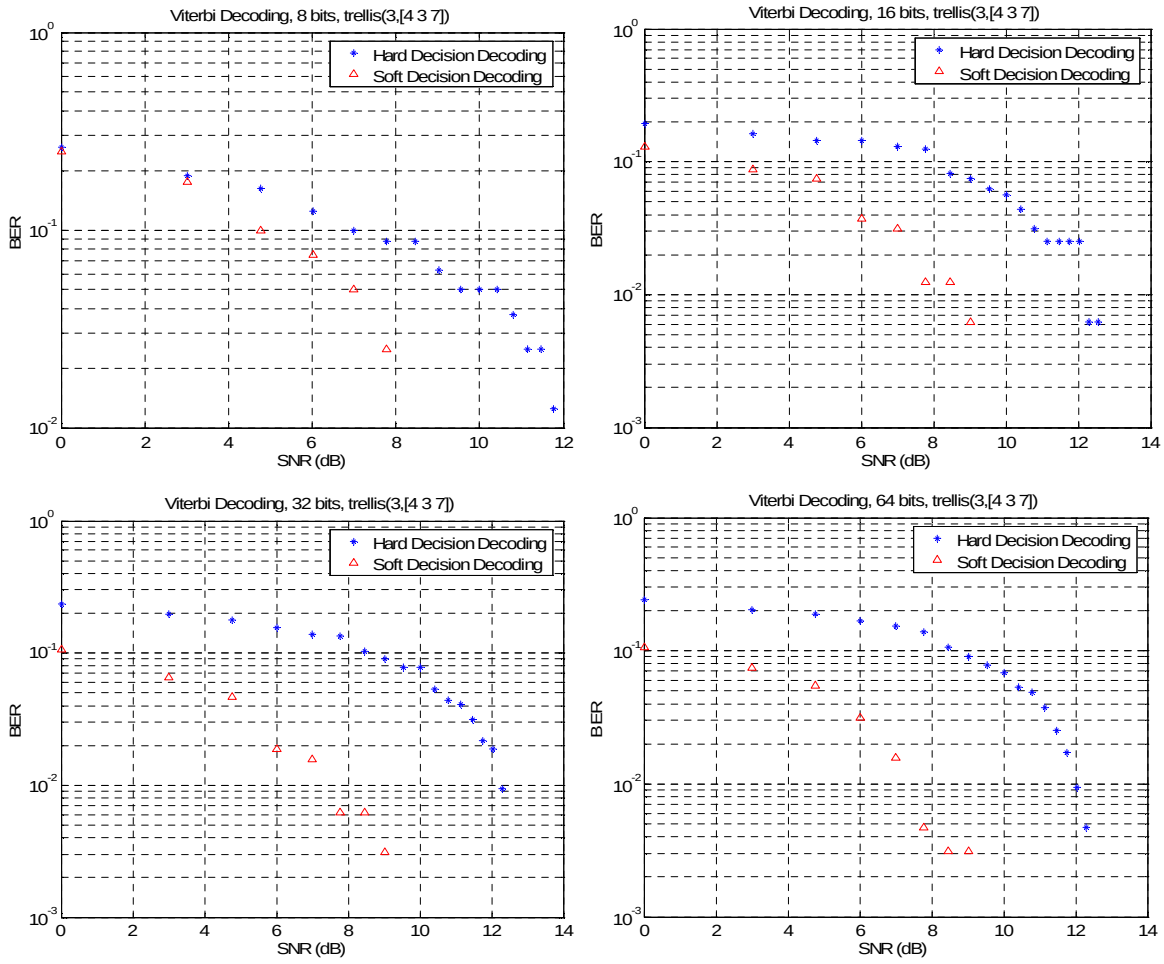


Figure 9.11 Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with logistic chaos encryption (μ=4) for a 1/3-rate (3, [4 3 7]) encoder for different message block size.

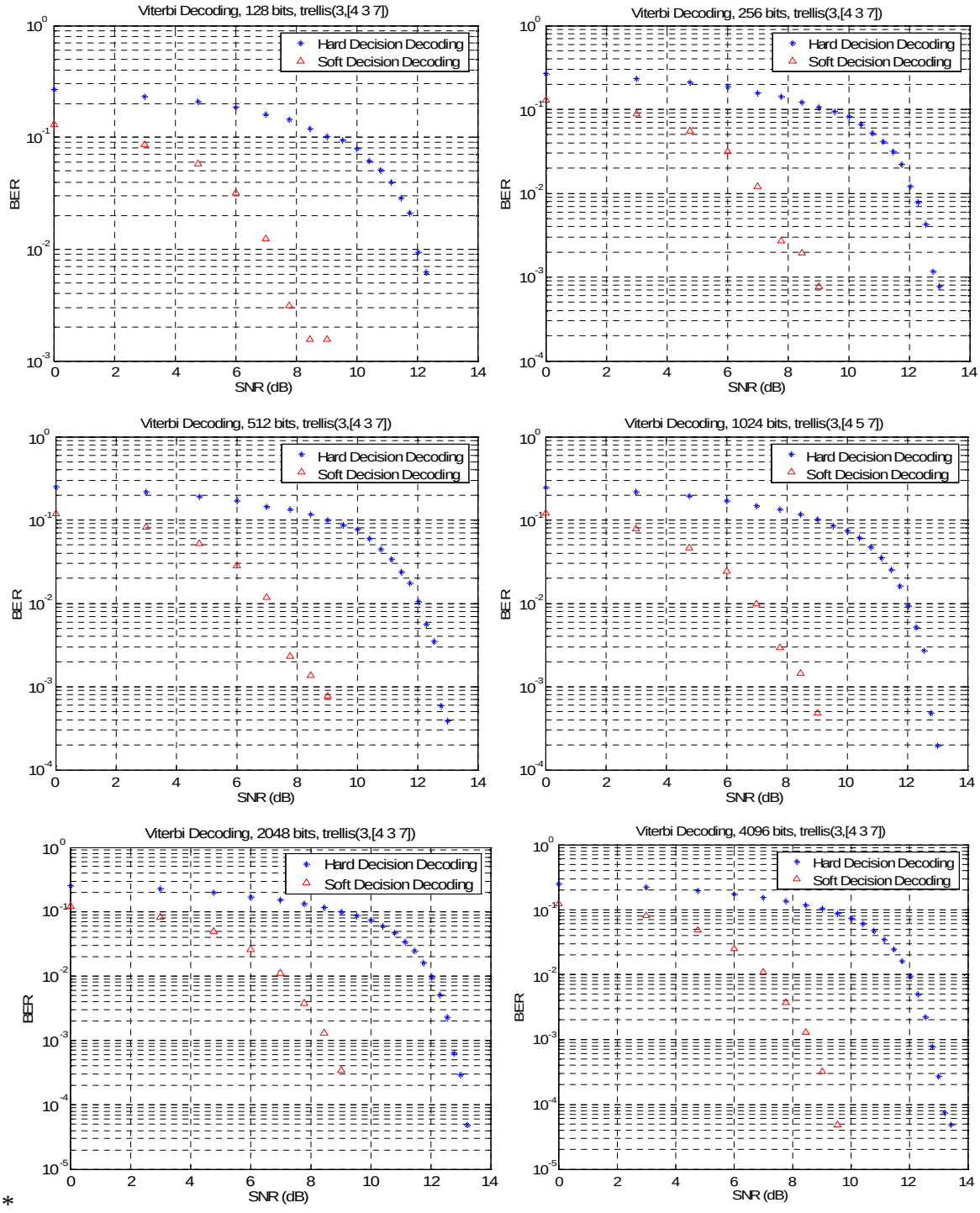


Figure 9.11, cont. Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with logistic chaos encryption (μ=4) for a 1/3-rate (3, [4 3 7]) encoder for different message block size.

9. 11.2 Algorithmic Complexity of the Multi-Trellis Code with Logistic System

To test the algorithmic complexity as well as bit error rate (BER) performance of the proposed method on multiple encoders, a MATLAB software model is built. The MATLAB model has the following inputs: the constraint length *K*, the rate, the generator polynomials, the decoding length *L*, and the SNR (E_b/N_0) in dB. The SNR value is varied from 1 dB to 100 dB with step size increment of one dB. Selected encoders are code1 (5,[37 33]); code2 (3,[4 3 7]); code3 (7,[171 133]); code4 ([4 3],[4 5 17;7 4 2]) and code5 ([5 4],[23 35 0;0 5 13]). Then the plots of the BER for the Viterbi decoder in a channel with Additive White Gaussian Noise (AWGN) are depicted. The results of algorithmic complexity analysis are summarized in Table 9.1. The LZ complexity analysis results of the selected character size for different rates of convolutional encoders are shown in Figure 9.12. As expected, all codes have exponential increase with character bit size, but code 3 has the most complex response to the input data and the code 1 has the least complex response. Note the true size of data for rate 1/3 is three times the original character size in bits. That is the reason why the algorithmic complexity of the original character is less than the complexity of series 4 and 5 codes. Also complexity does not perform very well on character sizes less than 64 bits. The code performance in the range of 64 to 256 bit character sizes are in an acceptable range, while 512 to 2048 bits demonstrate the optimum complexity class for the given set of encoders. In addition, code series 4 has the best output for 2048 bit character size with a complexity growth of 721 units from 1024 bit character size. Note the original message has been bit stuffed to make the message longer in size. In the case of a real data stream the variety will be higher; hence the encoder output response will have higher complexity. For this trial, the system promptly achieves the desired goal.

Table 9.1 Various rate convolutional codes and their complexity measures for selected character size data.

Encoder	**LZ Complexity**								
Character Series 1: Size (bits)	2048	1024	512	256	128	64	32	16	8
$C^{(1/2)}_2$(5,[37 33])	1250	1246	687	386	218	125	73	43	25
$C^{(1/3)}_3$(3,[4 3 7])	1098	1040	582	323	188	111	68	42	26
$C^{(1/2)}_4$(7,[171 133])	2138	1417	766	423	231	129	71	40	25
$C^{(2/3)}_5$([4 3],[4 5 17;7 4 2])	2102	1130	610	334	184	105	59	34	20
$C^{(2/3)}_6$([5 4],[23 35 0;0 5 13])	1467	1137	617	331	184	103	59	33	21

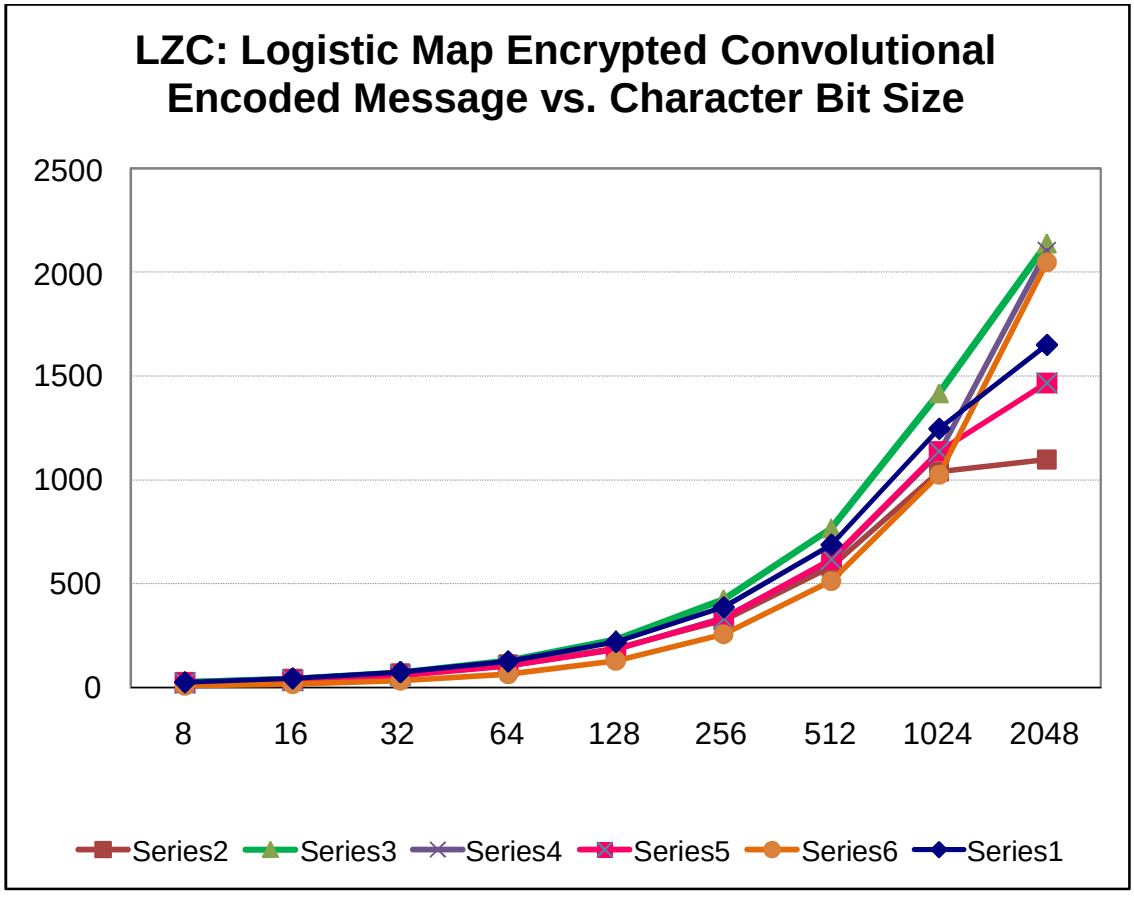


Figure 9.12 Viterbi decoder LZ complexity performance on AWGN channel with logistic chaos encryption (μ = 4) versus different message block size. Codes $C^{(1/2)}_2$(5,[37 33]), $C^{(1/3)}_3$(3,[4 3 7]) $C^{(1/2)}_4$(7,[171 133]), $C^{(2/3)}_5$([4 3],[4 5 17;7 4 2]), $C^{(2/3)}_6$([5 4],[23 35 0;0 5 13]).

Next, to illuminate the effect a prior data has on the encoder performance, two data maps are computed and analyzed. This trial utilizes the logistic chaos encrypted code at μ=4 with 1/3-rate on a (3 [4 5 7]) encoder for 512 by 512 bit selected images of a fish and a cameraman. Figure 9.13 and 9.14 show the original fish and cameraman images, their crypto-coded transformation image with logistic chaos encryption (μ=4) at 1/3-rate (3, [4 5 7]) encoder, the soft-decision decoded and hard-decision decoded-decrypted images and the plot of the soft decision and hard decision Viterbi decoder BER performance on an AWGN channel. Again a soft decision decoder has a much better bit error performance. The decoder is able to recover the encrypted data and is able to reasonably estimate the bit error rate for different signal strengths under a noisy AWGN channel. This indicates that the encoder can update the information map in real time to compensate for a changing data. Hence, occasional changes in the data stream could be accounted for in real time.

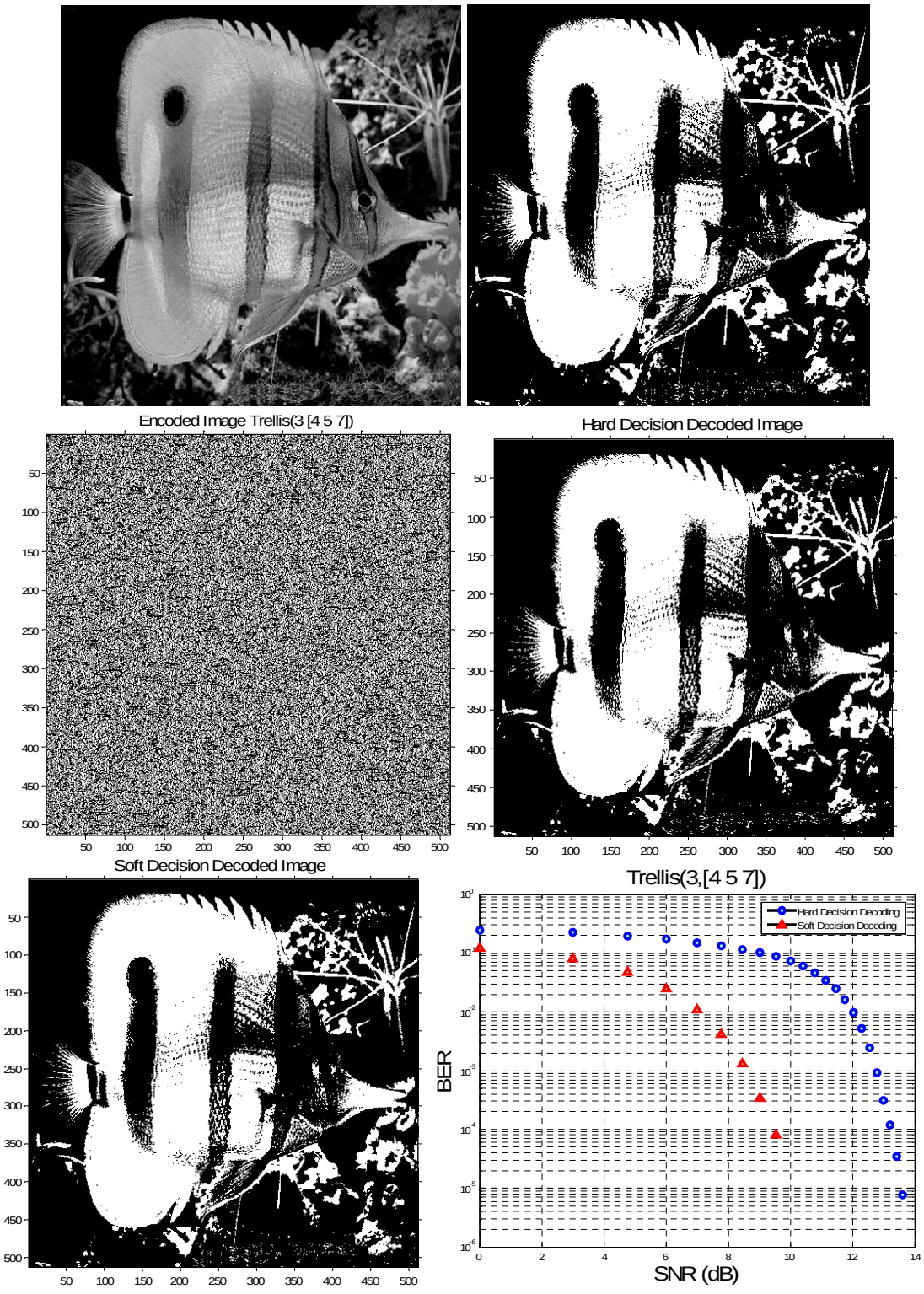


Figure 9.13 Soft decision and hard decision Viterbi decoder bit error rate performance on AWGN channel with logistic chaos encryption (μ=4) for 1/3-rate (3, [4 5 7]) encoder for fish image.

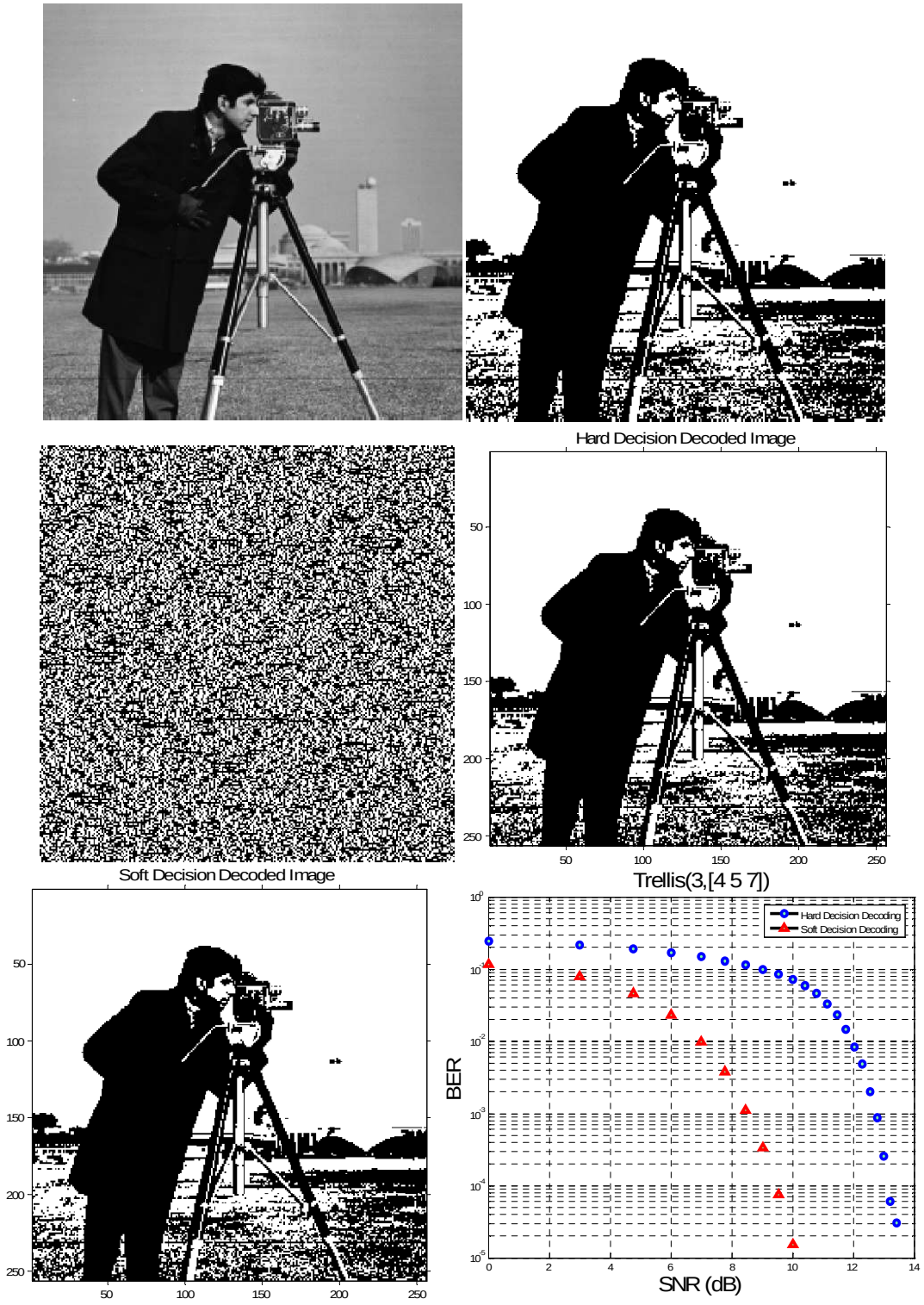


Figure 9.14 Soft decision and hard decision Viterbi decoder bit error rate performance on AWGN channel with logistic chaos encryption (μ = 4) for 1/3-rate (3, [4 5 7]) encoder for cameraman image.

9. 12 Algorithmic Complexity Analysis of Convolutionally Encoded Chaotic Encrypted Images

To compare and understand the intrinsic complexity performance of convolutional encoders, algorithmic complexity is computed and discussed. We apply chaotic encryption on the fish image using Logistic map as a mask generator with μ = 3.89. To simulate the channel noise the signal-to-noise ratio is varied from 1 dB to 100 dB with increments of 1 dB. Selected encoders are rate-1/3 code with constraint length=3 and generator polynomial = [4, 5, 7]; rate-1/2 code with constraint length=7 and generator polynomial = [171, 133]; rate-1/2 code with constraint length=5 and generator polynomial = [37 33]; rate-2/3 code with constraint length= [4, 3] and generator polynomial = [4 5 17; 7 4 2]; rate-2/3 code with constraint length=[5, 4] and generator polynomial = [23 35 0; 0 5 13]; rate-1/2 code with constraint length=3 and generator polynomial = [6, 7]; rate-1/3 code with constraint length=3 and generator polynomial = [3, 4, 5]; rate-1/2 code with constraint length=3 and generator polynomial = [7, 6] (5,[37 33]). Figure 9.15 shows the soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel.

We categorize properties of each encoder from the results summarized in Table 9.2. Simulation results compare 8 different encoders across three trials with different sequence lengths. The second, third and fourth encoders (code 2, code 3, code 4) have the most complex responses to the input data and the codes 1 and 7 are among the least complex performers.

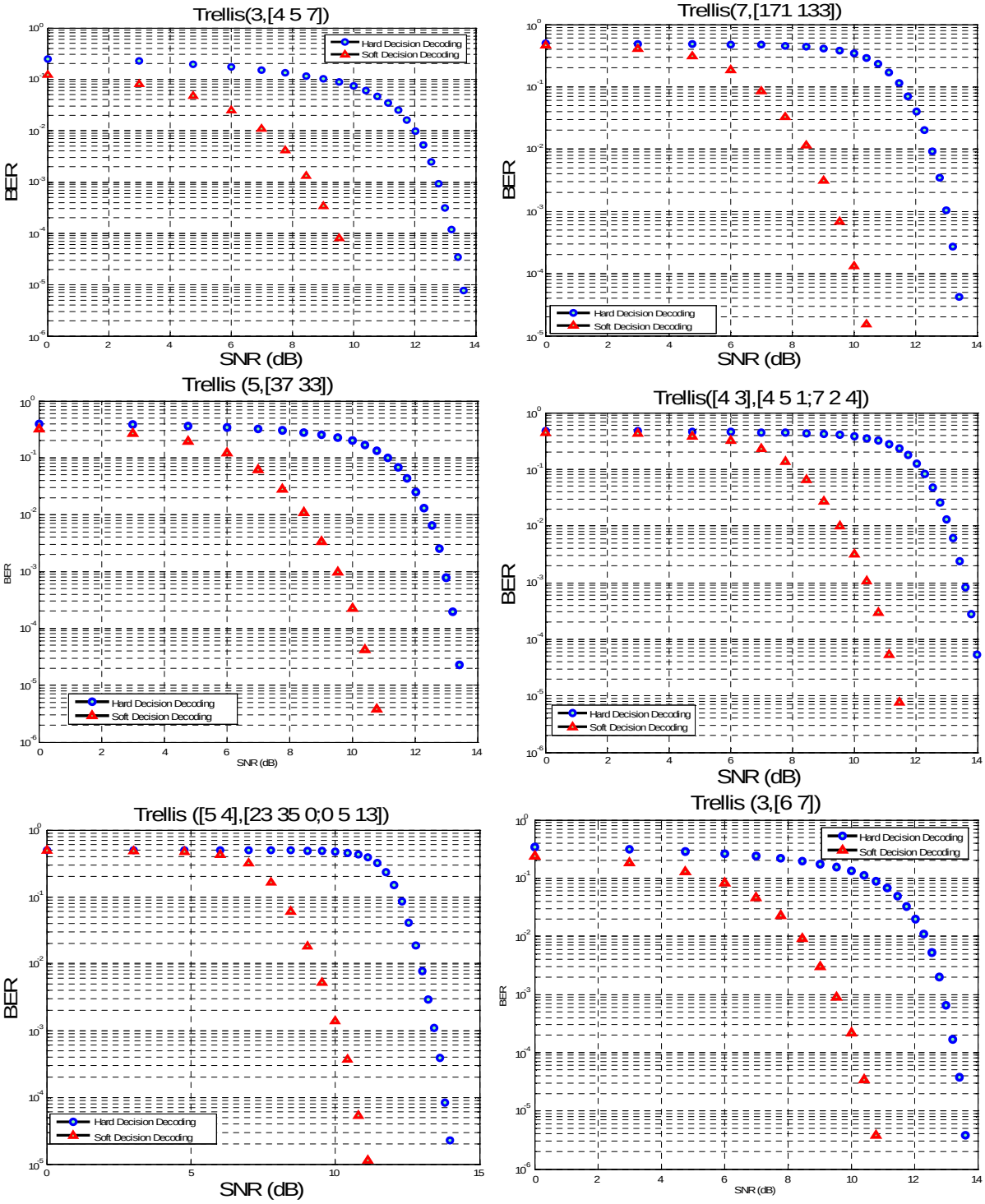


Figure 9.15 Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with logistic chaos encryption (μ = 4) for different rate encoders.

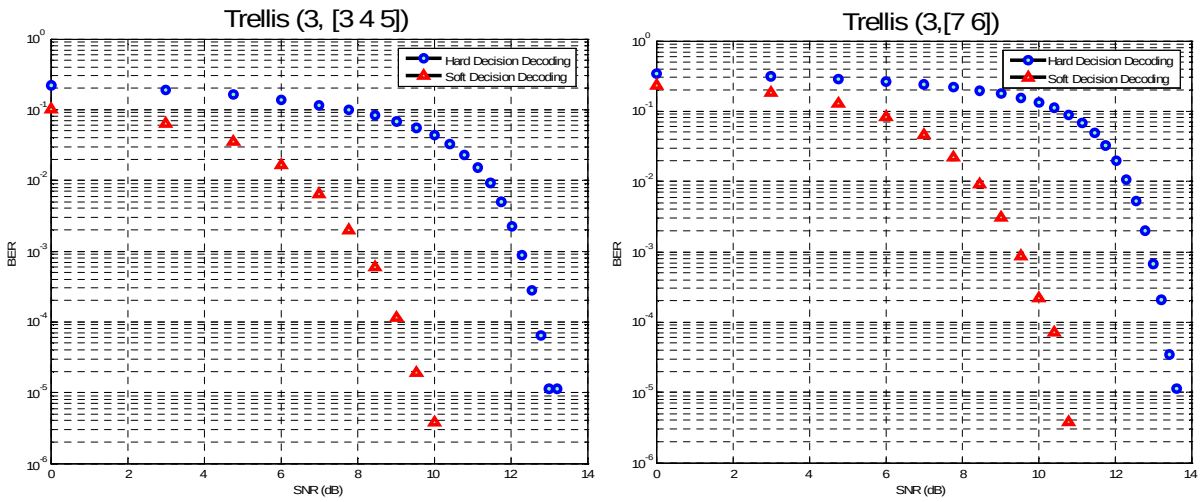


Figure 9.15, cont. Soft decision and hard decision Viterbi decoder bit error rate performance on AWGN channel with logistic chaos encryption (μ = 4) for different rate encoders.

Table 9.2 Various rate convolutional codes and their complexity measures for image data.

Encoder	**$L_1(S)$**	**$LZC_N \mid_{L1(S)}$**	**$L_2(S)$**	**$LZC_N \mid_{L2(S)}$**	**$L_3(S)$**	**$LZC_N \mid_{L3(S)}$**
C_1 (3, [4, 3, 7])	1099	0.504	806	0.513	494	0.530
C_2 (7, [171, 133])	2137	0.98	1555	0.991	929	0.997
C_3 (5, [37 33])	2208	1	1596	1	950	1
C_4 ([4, 3], [4 5 17; 7 4 2])	2229	1	1606	1	957	1
C_5 ([5, 4], [23 35 0; 0 5 13])	1867	0.856	1368	0.872	833	0.894
C_6 (3, [6, 7])	1546	0.709	1128	0.719	687	0.737
C_7 (3, [3, 4, 5])	1095	0.502	804	0.512	492	0.528
C_8 (3, [7, 6])	1412	0.647	1025	0.653	622	0.667

In calculating $C_N(S)$ we propose the following equation:

$$C\ (S) = C_0(S) + \kappa\ \Delta_{AVE} C(S), \tag{9.14}$$

where $\kappa = \left(\dfrac{n - 32700}{10000} \right)$, and $C_0(S)$ is the calculated LZ complexity for an array of length 32700 bits. Additionally, the normalized LZ complexity is computed from:

$$C_N(S) = \frac{C(S)}{\left(\frac{n}{\log_2(n)}\right)} = \frac{C_0(S) + \kappa\, \Delta_{AVE} C(S)}{\left(\frac{n}{\log_2(n)}\right)}. \tag{9.15}$$

To decrease the error in our estimate of normalized asymptotic LZ complexity we practically propose the half-rate increase. Therefore:

$$C_N(S) = \frac{C(S)}{\left(\frac{n}{\log_2(n)}\right)} = \frac{C_0(S) + \kappa \frac{\Delta_{AVE} C(S)}{2}}{\left(\frac{n}{\log_2(n)}\right)}. \tag{9.16}$$

The last row of Table 9.3 is calculated based on Eq. 9.16 for n=262000, notice that here n_0=32700.

$$C_N(S) = \frac{C_0(S) + \kappa \frac{\Delta_{AVE} C(S)}{2}}{\left(\frac{n}{\log_2(n)}\right)} = \frac{\left(C_0(S) + (22.93)\frac{\Delta_{AVE} C(S)}{2}\right)}{14556.2}. \tag{9.17}$$

While for string S_0, the normalized complexity is given by:

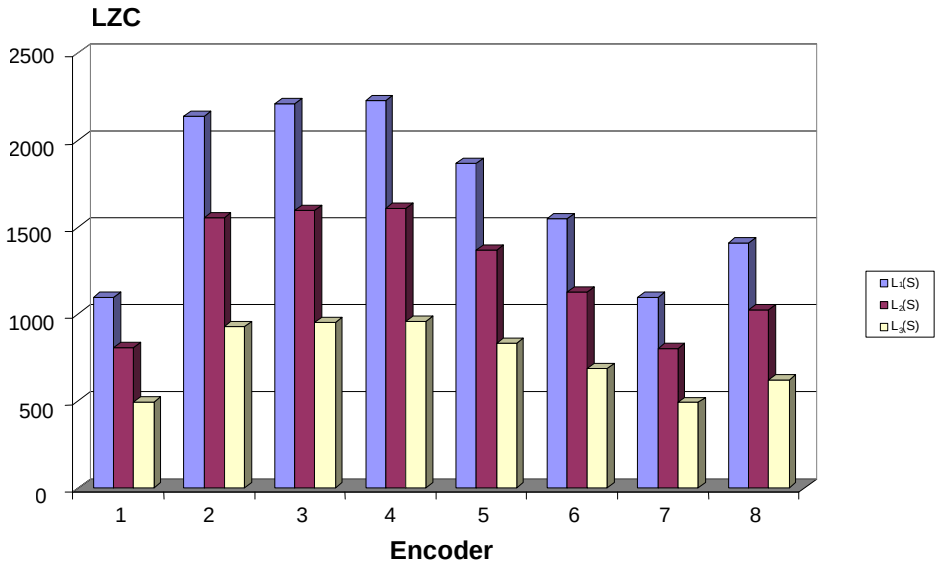
$$C_N(S_0) = \frac{C(S_0)}{\left(\frac{n_0}{\log_2(n_0)}\right)} = \frac{C(S_0)}{2180.43}. \tag{9.18}$$

Figure 9.16 shows the LZ complexity for lengths L_1, L_2 and L_3 and normalized LZ complexity performance with logistic chaos encryption (μ = 4) for given encoders. Eq. 9.17 is used in calculating the normalized LZC. We see that $C(S)$ is apparently an increasing function of L(S) = n considering each encoder. An interesting phenomenon is that for some encoders, like C_3 and C_4, $\Delta C(S)$ remains approximately constant. From Table 9.3 it is observed that $\Delta C(S)$ appears to remain almost constant on all scales in $L_j(S)$. This verifies that our conjecture in Eq. 9.14 precisely mirrors the performance of Lempel-Ziv complexity for large data sequences. This

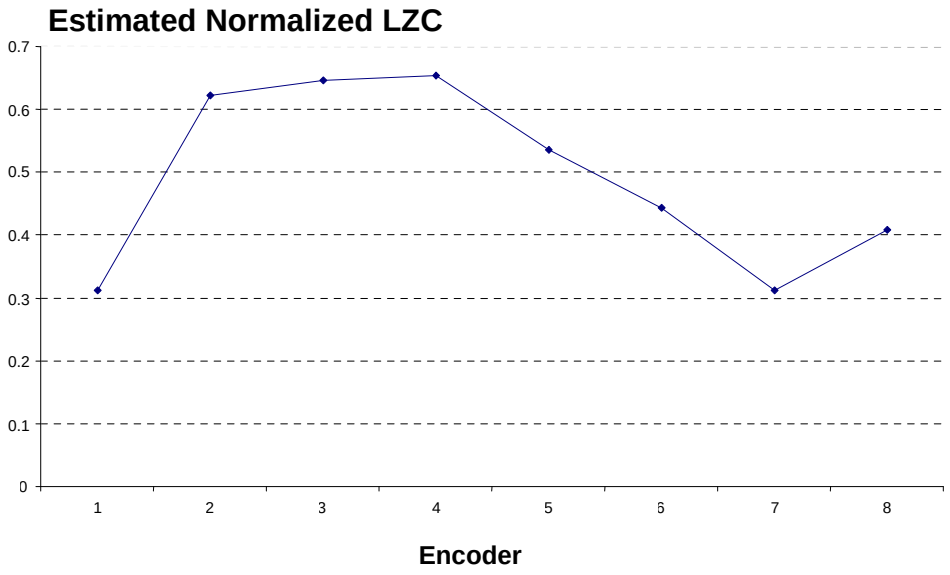
approach solves the problem of the buffer size limit and its cost in computing LZ complexity when a large amount of data is involved. In addition to that, the processing time is reduced using our method.

Table 9.3 Various rate convolutional codes and their complexity measures for image data.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
$L_1(S)$	1099	2137	2208	2229	1867	1546	1095	1412
$L_2(S)$	806	1555	1596	1606	1368	1128	804	1025
$L_3(S)$	494	929	950	957	833	687	492	622
ΔLZC_{AVE} per 10^3	29	58.2	61.3	62.3	50.7	41.8	29.1	38.7
ΔLZC_{AVE} per 10^4	301.5	604	629	636	517	429.5	301.5	395
$LZC_N(S_0)$	0.504	0.980	1	1	0.856	0.709	0.502	0.647
Lim $LZC_N(S)$ $S\rightarrow$large	0.313	0.622	0.647	0.654	0.535	0.444	0.313	0.408



(a)



(b)

Figure 9.16 Viterbi decoder (a) LZ complexity for lengths L_1, L_2 and L_3 (b) Normalized LZ complexity performance with logistic chaos encryption (μ = 4) for encoders: rate-1/3 code with constraint length=3 and G_1= [4, 5, 7]; rate-1/2 code with constraint length=7 and G_2= [171, 133]; rate-1/2 code with constraint length=5 and G_3= [37 33]; rate-2/3 code with constraint length= [4, 3] and G_4 = [4 5 17; 7 4 2]; rate-2/3 code with constraint length=[5, 4] and G_5 = [23 35 0; 0 5 13]; rate-1/2 code with constrain length=3 and G_6 = [6, 7]; rate-1/3 code with constraint length=3 and G_7 = [3, 4, 5]; and rate-1/2 code with constraint length=3 and G_8 = [7, 6] (5,[37 33]).

9. 13 Complexity Analysis of the Best Rate-1/2 Trellis Encoded Images

In this section a base LZ complexity of a logistic chaos encrypted data without a convolutional coding is calculated. This is denoted by LZC_{MNC} = 352 for a string of 4096 bits. The ΔLZC represents the complexity increase because of convolutionally coding of the selected test image (TEXT.gif file). This is computed by subtracting LZC_{MNC} from the values of first column in Table 9.5. These results are shown in second column. Third column contains the LZC values of the convolutionally encoded text image without chaotic masking. Since these are the best rate-1/2 encoders, the data size at the encoder output is twice the original data size, i.e., 8192 bits. The best rate-1/2 convolutional encoders are: $Code_1$(3,[7 5]); $Code_2$(4,[17 15]); $Code_3$(5,[35 23]); $Code_4$(6,[75 53]); $Code_5$(7,[171 133]); $Code_6$(8,[371 247]); $Code_7$(9, [753 561); $Code_8$(10,[1545 1167]); $Code_9$(11,[3661 2335]); $Code_{10}$(12,[5723 4335]); $Code_{11}$ (13,[17661 10533]); and $Code_{12}$ (14,[27123 21675]). Figure 9.18 shows the best rate-½ convolutional encoded and Viterbi decoded for the logistic chaos encrypted text image. Figure 9.19 demonstrates the soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with Logistic chaos encryption (μ=4) for the best rate-½ convolutional codes. The complexity measure analysis is applied to three types of data sequences. These sequences are the logistic chaos Masked and Convolutionally Coded (MC) sequence, the logistic chaos Masked No Coding (MNC) sequence and the convolutionally Coded Not Masked (CNM) sequence. The corresponding LZ complexities are LZC_{MC}, LZC_{MNC}, LZC_{CNM}. $\Delta LZC = LZC_{MC} - LZC_{MNC}$ indicates the effect of the complexity increase because of convolutional coding. Figure 9.19 shows the complexity computation result plot. Free distance and bounded free distance values for each encoder are given in Figure 9.20. The simulations have confirmed that both chaotic encryption and convolutional encoding design have significant combined effect on the

system output complexity. However, convolutional coding alone does not provide high level of algorithmic complexity in the coding process output. This shows that the components of the unified crypto-encoder system architecture described in this trial have proved to be beneficial for cryptosystem operating in realistic simulations when small packets of data sequences are involved.

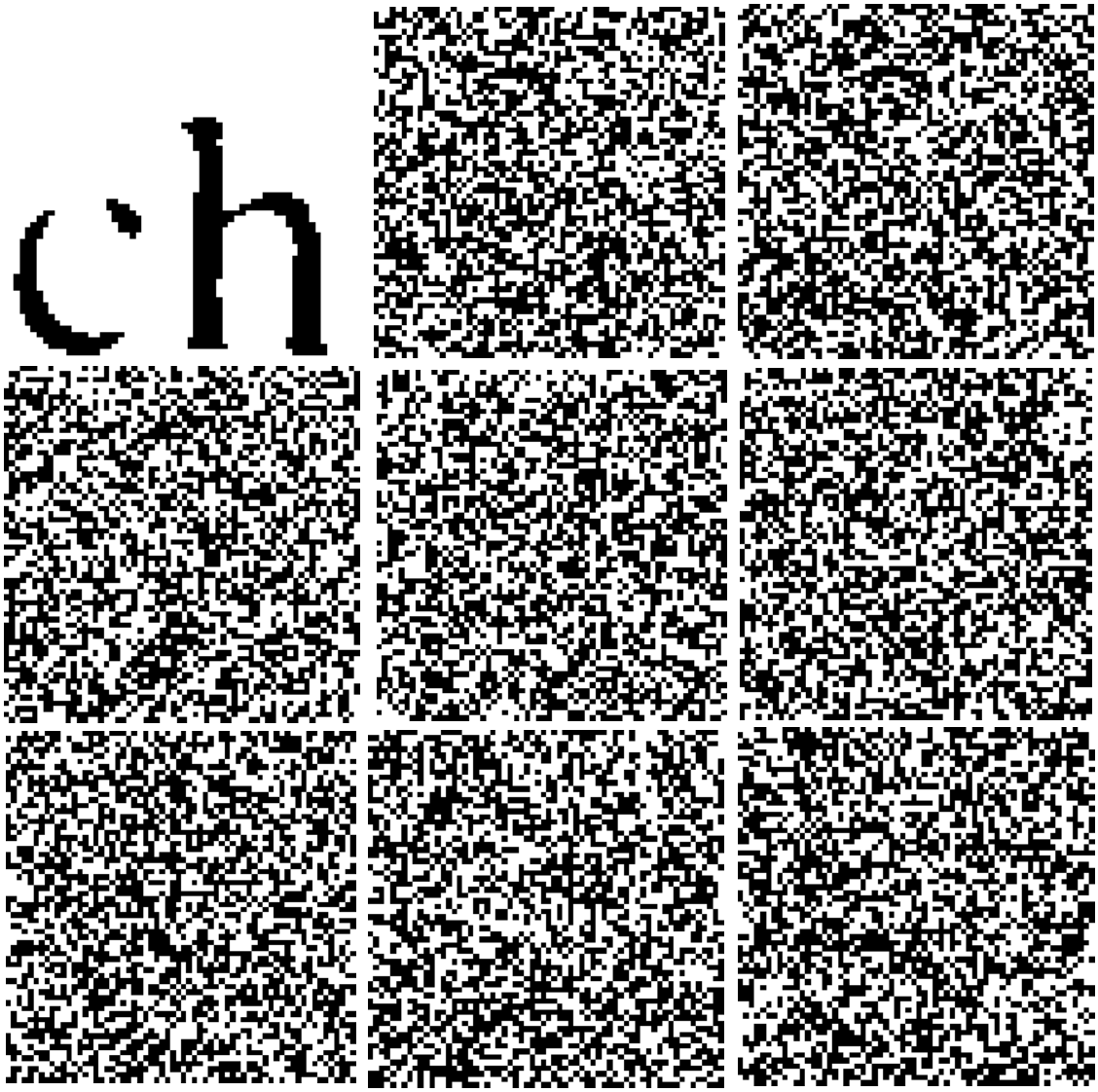


Figure 9.17 Best rate-½ convolutional encoded and Viterbi decoded for logistic chaos encrypted text image.

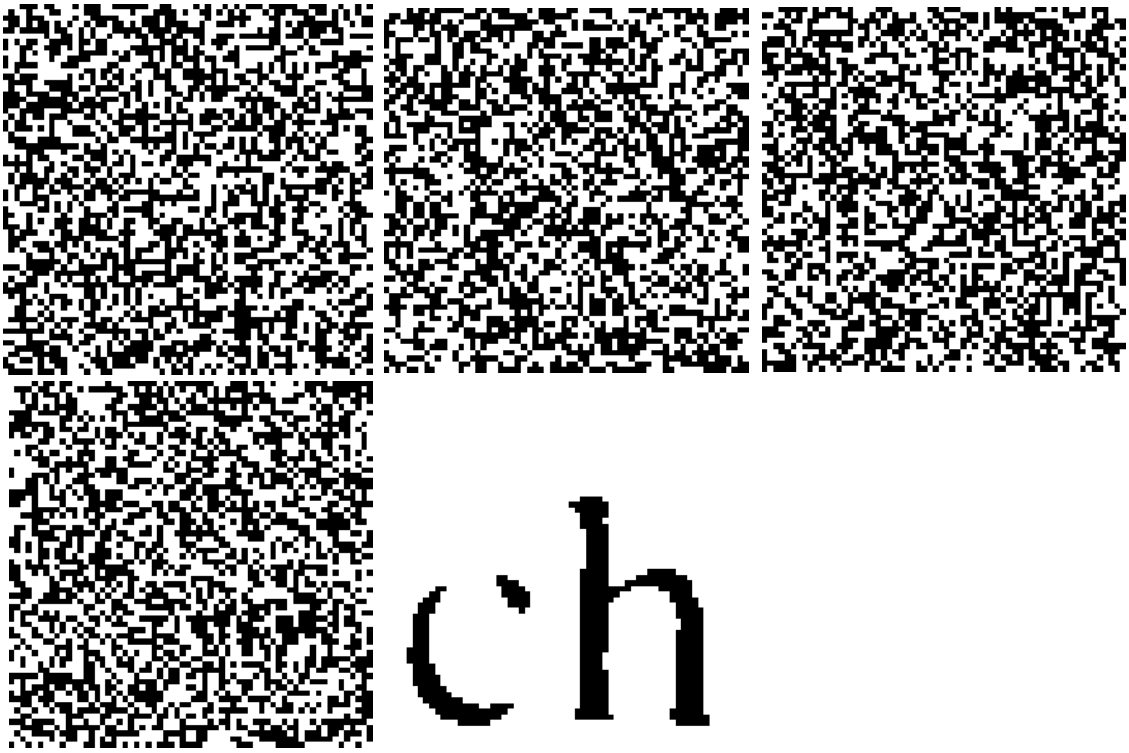


Figure 9.17, cont. Best rate-½ convolutional encoded and Viterbi decoded for logistic chaos encrypted text image.

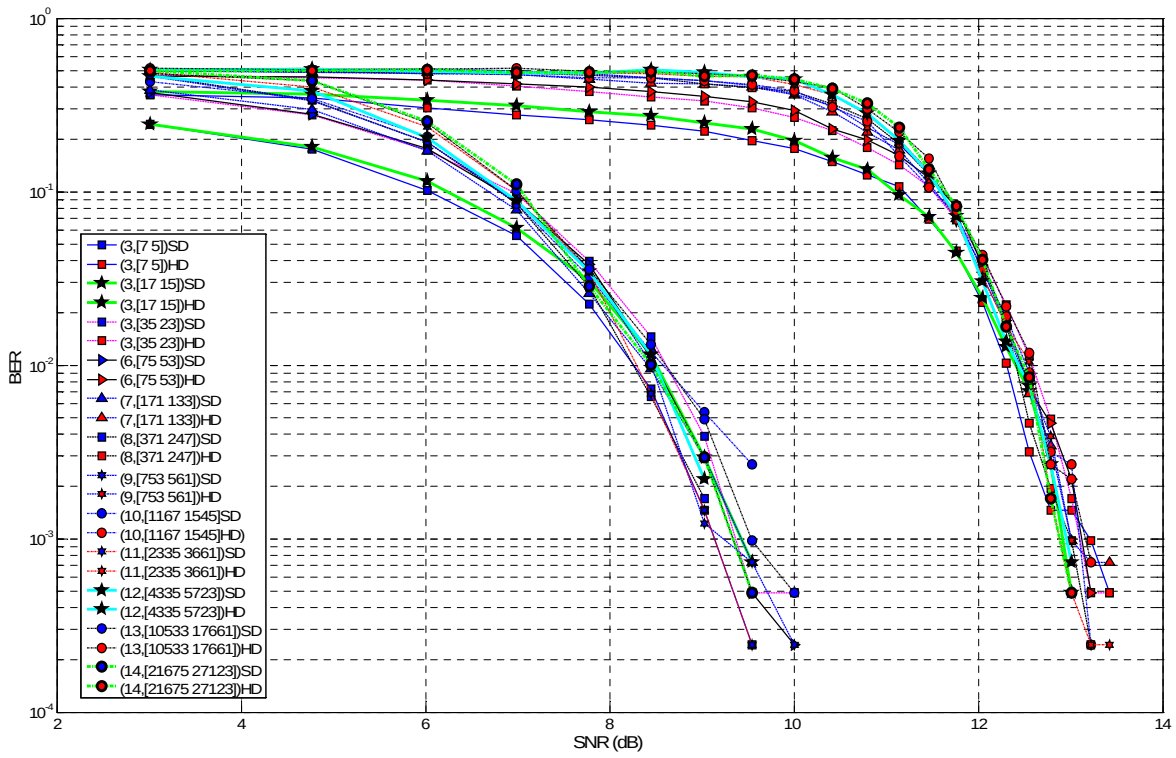


Figure 9.18 Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with Logistic chaos encryption (μ=4) for best rate-½ convolutional codes: $Code_1$(3,[7 5]); $Code_2$(4,[17 15]); $Code_3$(5,[35 23]); $Code_4$(6,[75 53]); $Code_5$(7,[171 133]); $Code_6$(8,[371 247]); $Code_7$(9, [753 561); $Code_8$(10,[1545 1167]); $Code_9$(11,[3661 2335]); $Code_{10}$(12,[5723 4335]); $Code_{11}$ (13,[17661 10533]); and $Code_{12}$ (14,[27123 21675]).

Table 9.4 Best rate-1/2 max free distance convolutional codes, and their complexity measures for text size (64 by 64 bits).

Encoder Trellis	LZC_{MC} Masked and Coded	$\Delta LZC = LZC_{MC} - LZC_{MNC}$ increase because of Coding	LZC_{CNM} Coded No Mask	d_{free}	d_{free} upper bound
$C_1^{(1/2)}$(3, [7 5])	474	122	40	5	5
$C_2^{(1/2)}$(4, [17 15])	526	174	40	6	6
$C_3^{(1/2)}$(5, [35 23])	568	216	41	7	8
$C_4^{(1/2)}$(6, [75 53])	610	258	43	8	8
$C_5^{(1/2)}$(7, [171 133])	634	282	47	10	10
$C_6^{(1/2)}$(8, [371 247])	645	293	49	10	11
$C_7^{(1/2)}$(9, [753 561])	649	297	54	12	12
$C_8^{(1/2)}$(10, ,[1545 1167])	642	290	54	12	13
$C_9^{(1/2)}$(11,[3661 2335])	652	300	58	14	14
$C_{10}^{(1/2)}$(12,[5723 4335])	649	297	64	15	15
$C_{11}^{(1/2)}$(13,[17661 10533])	652	300	69	16	16
$C_{12}^{(1/2)}$(14,[27123 21675])	649	297	71	16	17

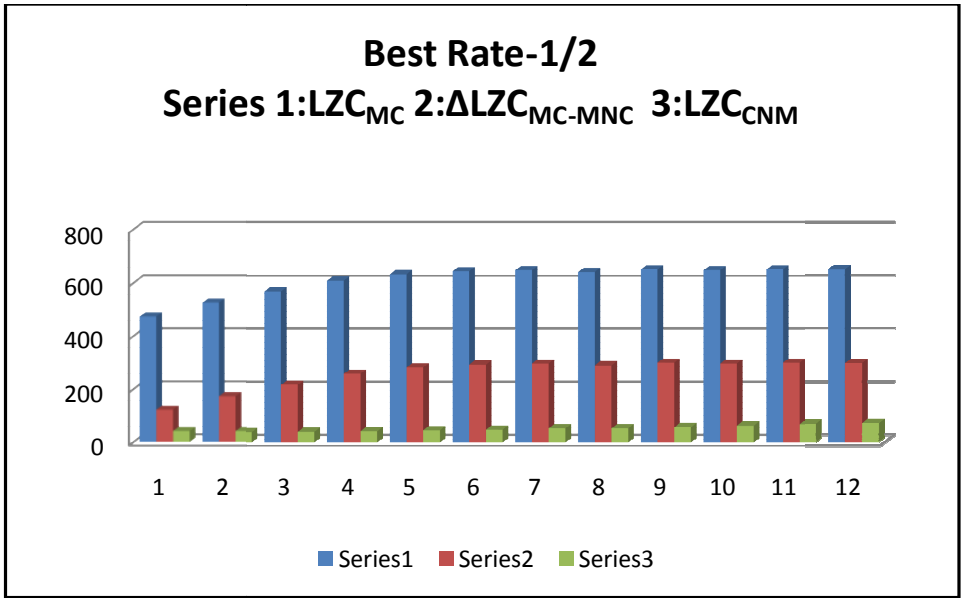


Figure 9.19 LZC for logistic chaos Masked and best rate-1/2 convolutionally Coded (MC) text; logistic chaos Masked No Coding (MNC); convolutionally Coded No chaos Mask (CNM).

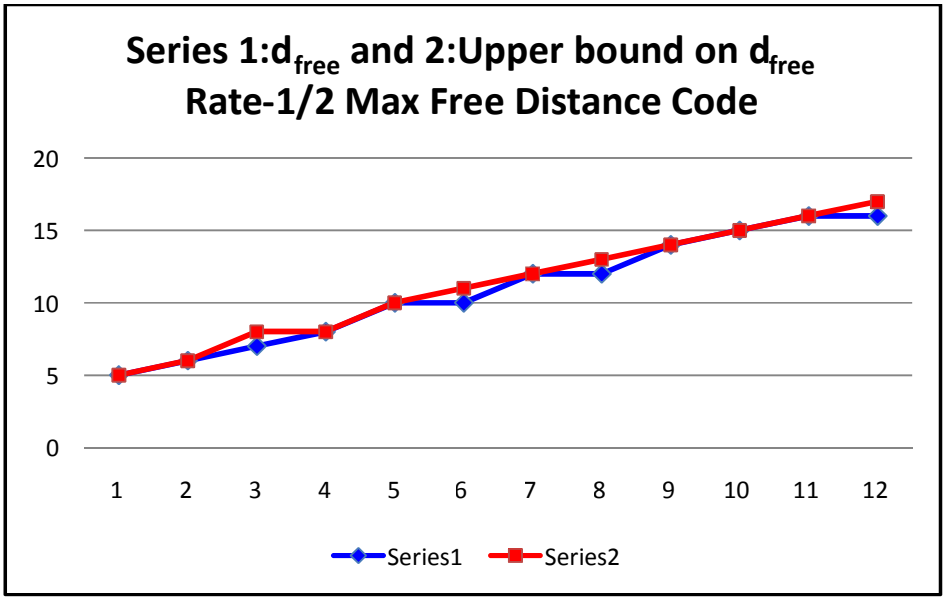


Figure 9.20 d_{free} (series1) and upper bound on d_{free} (series2) for rate-1/2 max free distance codes as described in Table 9.4.

9. 14 Rate-1/*n* Maximum Free Distance Convolutional Codes: Algorithmic Complexity with Chaotic-Encrypted Data

This section introduces the rate-1/*n* maximum free distance convolutional codes and their algorithmic complexity metrics. This is done to determine the performance of these encoders with chaotic-encrypted data. Here we present a measure of an encoder's ability to provide confusion while reducing the error expectation. Information in this context can be understood by referring back to coding theory. Recall that convolutional code provides an error correcting estimate that goes as free distance. In particular, the free distance metric can be use to determine the error correcting estimation. This quantity encompasses all of the knowledge of the encoder. Hence, by observing the change in free distance during the perceptive phase of an encoding system, one can determine which encoder is most instrumental in producing better error correcting performance. The exact steps required to compute the LZ complexity of chaotic-encrypted and encoded information are given by following algorithm.

Table 9.5 LZ Complexity Computation Algorithm
1: compute the mask function ***2: apply mask function*** ***3: encode data*** ***4: for all SNR do*** ***5: compute BER*** ***6: for all $L_j(S)$ do*** ***7: compute C(S)*** ***8: compute ΔLZC_{AVE}*** ***9: estimate $LZC_N(S)$ when $S \rightarrow large$*** ***10: decode data***

The development of the information processing is done in four steps. First the best rate-1/*n* is selected. Then the encoder is defined. Next, the information sequence length is varied and complexity is computed for the selected sequence. Then, the average complexity increment per 10000 bits (ΔLZC_{AVE}) is computed. This is the quantity from which the normalized LZ complexity for large sequence length can be estimated. We notice that when simulations are implemented by algorithm in Table 9.5, they reveal the time complexity too. By working through this, one discovers that the time required by the basic arithmetic operations and the decoding logic is higher whenever codes with large free distance (higher LZC) are applied. We have tried the experiment on the rates 1/2 through 1/7 convolutional encoders with maximum free distance performance. A summary of each rate-1/*n* encoder system detail is provided in Tables 9.6 through 9.13 and these are followed by the complexity analysis and free distance diagrams.

Best Rate-1/2 Class

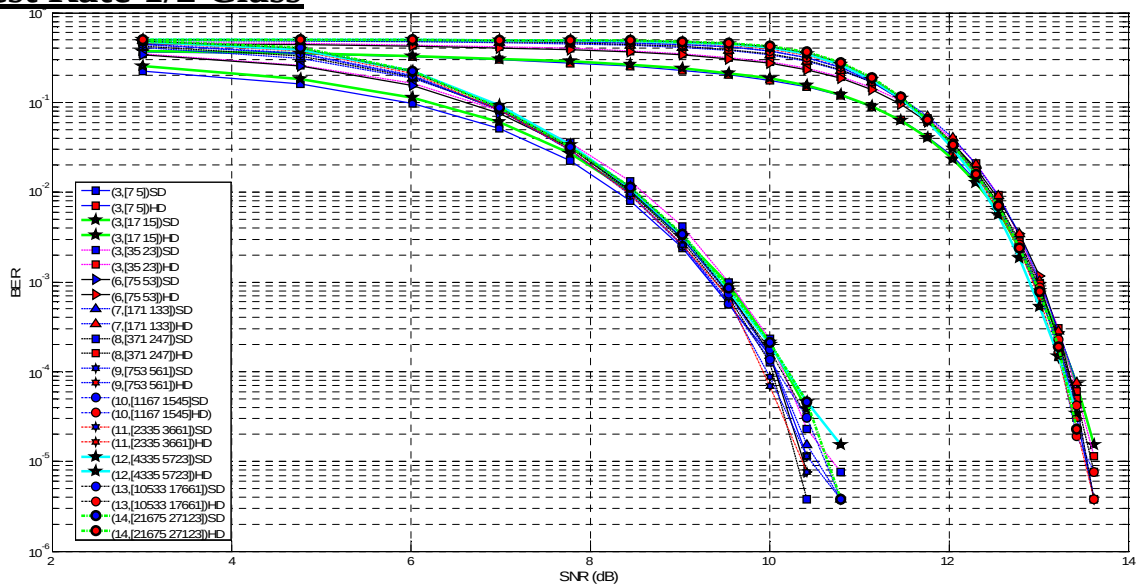


Figure 9.21 Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with Logistic chaos encryption (μ=4) for best rate-1/2 convolutional codes: $Code_1$(3,[7 5]); $Code_2$(4,[17 15]); $Code_3$(5,[35 23]); $Code_4$(6,[75 53]); $Code_5$(7,[171 133]); $Code_6$(8,[371 247]); $Code_7$(9, [753 561); $Code_8$(10,[1545 1167]); $Code_9$(11,[3661 2335]); $Code_{10}$(12,[5723 4335]); $Code_{11}$(13,[17661 10533]); and $Code_{12}$(14,[27123 21675]).

Table 9.6 Best rate-1/2 convolutional codes LZC and LZC_N

Encoder	**$LZC\|_{L1(S)}$**	**b(n)= 2180.4 LZC_N**	**$LZC\|_{L2(S)}$**	**b(n)= 1568.7 LZC_N**	**$LZC\|_{L3(S)}$**	**b(n)= 931.6 LZC_N**
$C_1^{(1/2)}$(3, [7 5])	1544	0.71	1127	0.722	687	0.741
$C_2^{(1/2)}$(4, [17 15])	1694	0.775	1240	0.788	758	0.809
$C_3^{(1/2)}$(5, [35 23])	1869	0.858	1370	0.872	839	0.897
$C_4^{(1/2)}$(6, [75 53])	2017	0.928	1474	0.940	890	0.956
$C_5^{(1/2)}$(7, [171 133])	2128	0.980	1547	0.991	933	0.997
$C_6^{(1/2)}$(8, [371 247])	2194	1	1589	1	945	1
$C_7^{(1/2)}$(9, [753 561])	2217	1	1599	1	955	1
$C_8^{(1/2)}$(10, ,[1545 1167])	2234	1	1611	1	962	1
$C_9^{(1/2)}$(11,[3661 2335])	2233	1	1611	1	960	1
$C_{10}^{(1/2)}$(12,[5723 4335])	2226	1	1604	1	953	1
$C_{11}^{(1/2)}$(13,[17661 10533])	2225	1	1607	1	956	1
$C_{12}^{(1/2)}$(14,[27123 21675])	2235	1	1611	1	958	1

Table 9.7 Best rate-1/2 convolutional codes and their complexity measures and weight structure

Rate-1/2	$C_1^{(1/2)}$	$C_2^{(1/2)}$	$C_3^{(1/2)}$	$C_4^{(1/2)}$	$C_5^{(1/2)}$	$C_6^{(1/2)}$	$C_7^{(1/2)}$	$C_8^{(1/2)}$	$C_9^{(1/2)}$	$C_{10}^{(1/2)}$	$C_{11}^{(1/2)}$	$C_{12}^{(1/2)}$
$\mathbf{LZC}\vert_{L1(S)}$	1544	1694	1869	2017	2128	2194	2217	2234	2233	2226	2225	2235
$\mathbf{LZC}\vert_{L2(S)}$	1127	1240	1370	1474	1547	1589	1604	1611	1611	1604	1607	1611
$\mathbf{LZC}\vert_{L3(S)}$	687	758	839	890	933	945	953	962	960	953	956	958
$\mathbf{d_{free}}$	5	6	7	8	10	10	12	12	14	15	16	16
$\mathbf{d_{free}\vert up}$	5	6	7	8	10	11	12	13	14	15	16	17
$\Delta\mathbf{LZC_{AVE}}$ **per** $\mathbf{10^4}$	417	454	499	543	581	605	613	623	622	622	618	624
Lim $\mathbf{LZC_N(S)}$ $\mathbf{S\rightarrow large}$	0.43	0.47	0.52	0.57	0.60	0.63	0.64	0.64	0.64	0.64	0.64	0.64

Figure 9.21 shows the soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel. We summarized the complexity analysis results in Tables 9.6 and 9.7 in which we compare 12 different encoders across three trials with different sequence lengths. Additionally, in Figures 9.22, 9.23 and 9.24 we have depicted the LZC, LZC_N and free distance performance of this trial with best rate-1/2 convolutionally encoded data. Accordingly, the least complex performer is $C_1^{(1/2)}$. Code $C_{12}^{(1/2)}$ has the most complex responses to the input data followed by codes $C_8^{(1/2)}$ and $C_9^{(1/2)}$ in second and third place by only one unit difference in LZC, respectively. Notice that despite the small differences in LZC there are remarkable differences in time complexity because of the constraint length difference between these codes which can account for better error correcting performance (larger free distance) in in the $C_{12}^{(1/2)}$ code. It noteworthy that free distance of $C_{12}^{(1/2)}$ and $C_{11}^{(1/2)}$; $C_8^{(1/2)}$ and $C_7^{(1/2)}$; $C_6^{(1/2)}$ and $C_5^{(1/2)}$

are equal although their LZCs are different. These facts are useful for applications where the time complexity is the dominating factor with respect to algorithmic complexity performance.

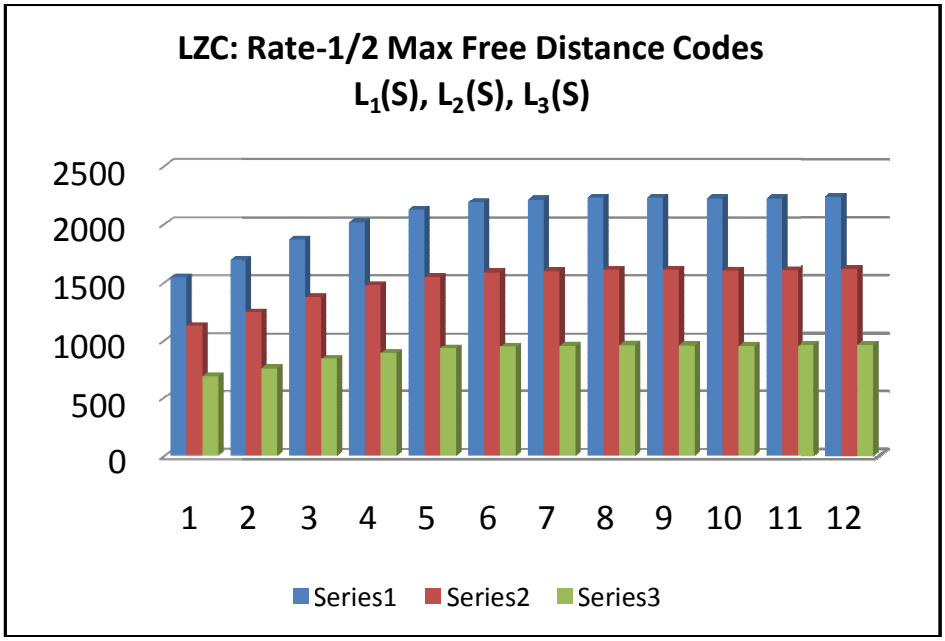


Figure 9.22 LZC for logistic chaos masked and best rate-1/2 convolutionally encoded.

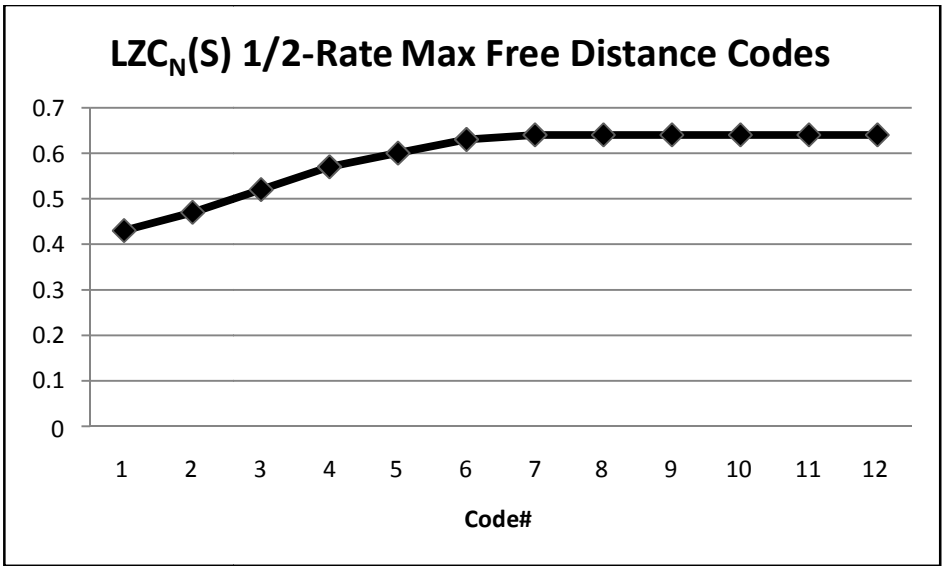


Figure 9.23 LZC_N for logistic chaos masked and best rate-1/2 convolutionally encoded.

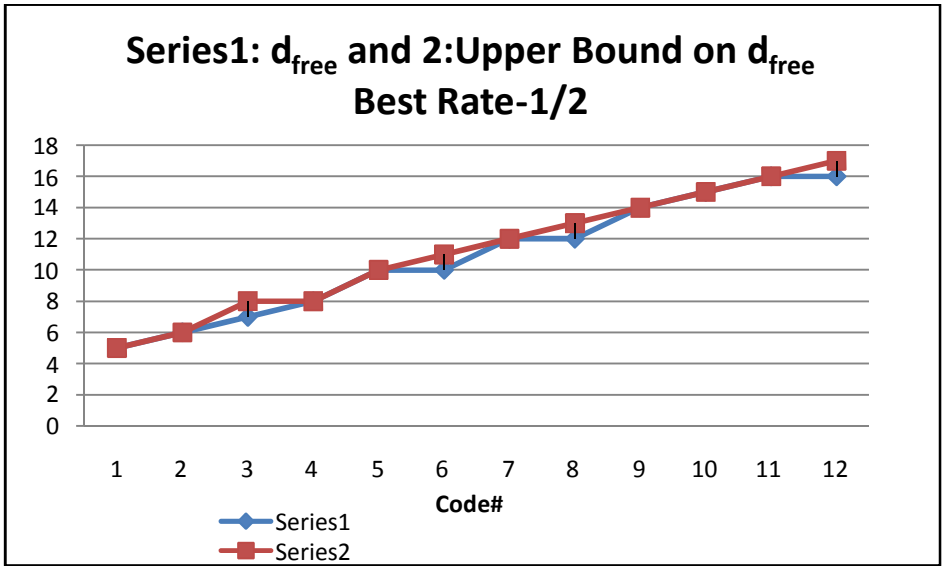


Figure 9.24 d_{free} (series1) and upper bound on d_{free} (series2) for best rate-1/2 maximum free distance code.

Best Rate-1/3 Class

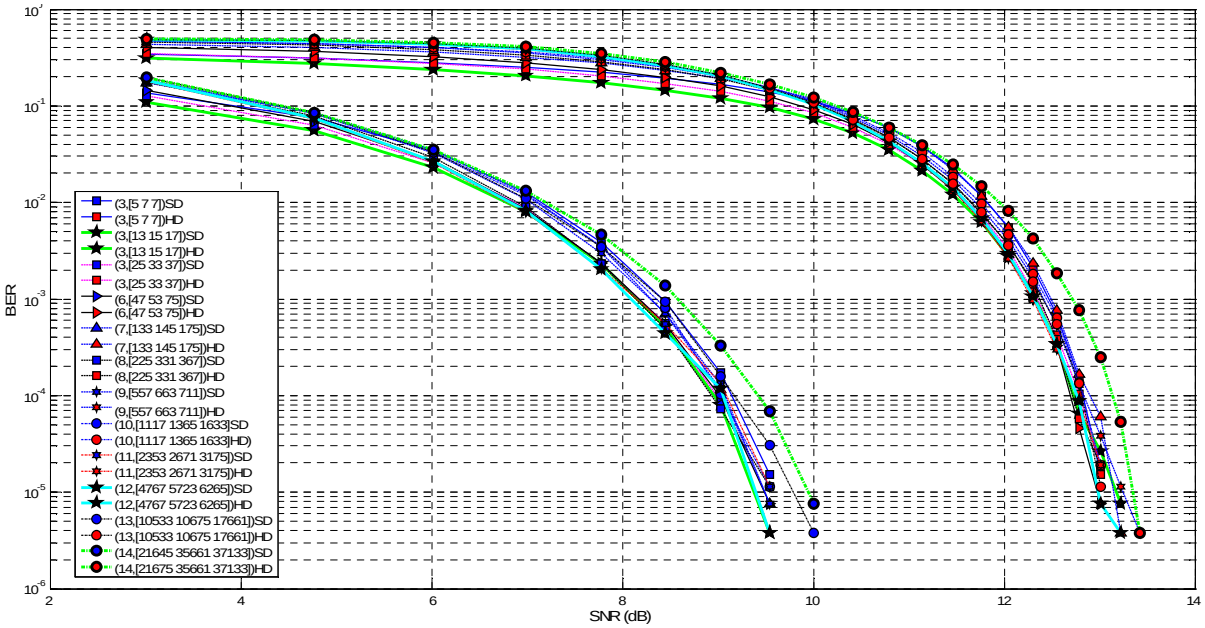


Figure 9.25 Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with logistic chaos encryption (μ = 4) for best rate-1/3 convolutional codes: $C_1^{(1/3)}$(3,[5 7 7]); $C_2^{(1/3)}$(4,[13 15 17]); $C_3^{(1/3)}$(5,[25 33 37]); $C_4^{(1/3)}$(6,[47 53 75]); $C_5^{(1/3)}$(7,[133 145 175]); $C_6^{(1/3)}$(8,[225 331 367]); $C_7^{(1/3)}$(9, [557 663 711]); $C_8^{(1/3)}$(10,[1117 1365 1633]); $C_9^{(1/3)}$(11,[2353 2671 3175]); $C_{10}^{(1/3)}$(12,[4767 5723 6265]); $C_{11}^{(1/3)}$(13,[10533 10675 17661]); and $C_{12}^{(1/3)}$(14,[21645 35661 37133]).

Table 9.8 Best rate-1/3 convolutional codes LZC and LZC_N

Encoder	**$LZC\|_{L1(S)}$**	**$b(L_1(S))=2180.4\ LZC_N$**	**$LZC\|_{L2(S)}$**	**$b(L_2(S))=1568.7\ LZC_N$**	**$LZC\|_{L3(S)}$**	**$b(L_3(S))=931.6\ LZC_N$**
$C_1^{(1/3)}$ (3,[5 7 7])	1097	0.503	806	0.513	494	0.53
$C_2^{(1/3)}$ (4,[13 15 17])	1214	0.556	895	0.570	558	0.598
$C_3^{(1/3)}$ (5, [25 33 37])	1368	0.627	1015	0.647	633	0.679
$C_4^{(1/3)}$ (6,[47 53 75])	1546	0.709	1145	0.729	719	0.771
$C_5^{(1/3)}$ (7,[133 145 175])	1748	0.801	1296	0.826	804	0.862
$C_6^{(1/3)}$ (8,[225 331 367])	1907	0.874	1401	0.893	857	0.919
$C_7^{(1/3)}$ (9, [557 663 711])	2076	0.952	1508	0.961	918	0.985
$C_8^{(1/3)}$ (10,[1117 1365 1633])	2151	0.986	1556	0.992	934	1
$C_9^{(1/3)}$ (11,[2353 2671 3175])	2154	0.988	1559	0.993	938	1
$C_{10}^{(1/3)}$ (12,[4767 5723 6265])	2216	1	1599	1	959	1
$C_{11}^{(1/3)}$ (13,[10533 10675 17661])	2221	1	1602	1	951	1
$C_{12}^{(1/3)}$ (14,[21645 35661 37133])	2234	1	1612	1	956	1

Table 9.9 Best rate-1/3 convolutional codes and their complexity measures and weight structure.

Rate-1/3	**C_1**	**C_2**	**C_3**	**C_4**	**C_5**	**C_6**	**C_7**	**C_8**	**C_9**	**C_{10}**	**C_{11}**	**C_{12}**
$LZC\|_{L1(S)}$	1097	1214	1368	1546	1748	1907	2076	2076	2154	2216	2221	2234
$LZC\|_{L2(S)}$	806	895	1015	1145	1296	1401	1508	1556	1559	1599	1602	1612
$LZC\|_{L3(S)}$	494	558	633	719	1401	857	918	934	938	959	951	956
d_{free}	8	10	12	13	15	16	18	20	22	24	24	26
d_{free}|up	8	10	12	13	15	16	18	20	22	24	24	26
Δ_{AVE} LZC per 10^4	291	319	353	401	452	506	568	595	595	617	619	622
Lim $LZC_N(S)$ S→large	0.304	0.334	0.372	0.422	0.476	0.529	0.590	0.616	0.616	0.638	0.640	0.643

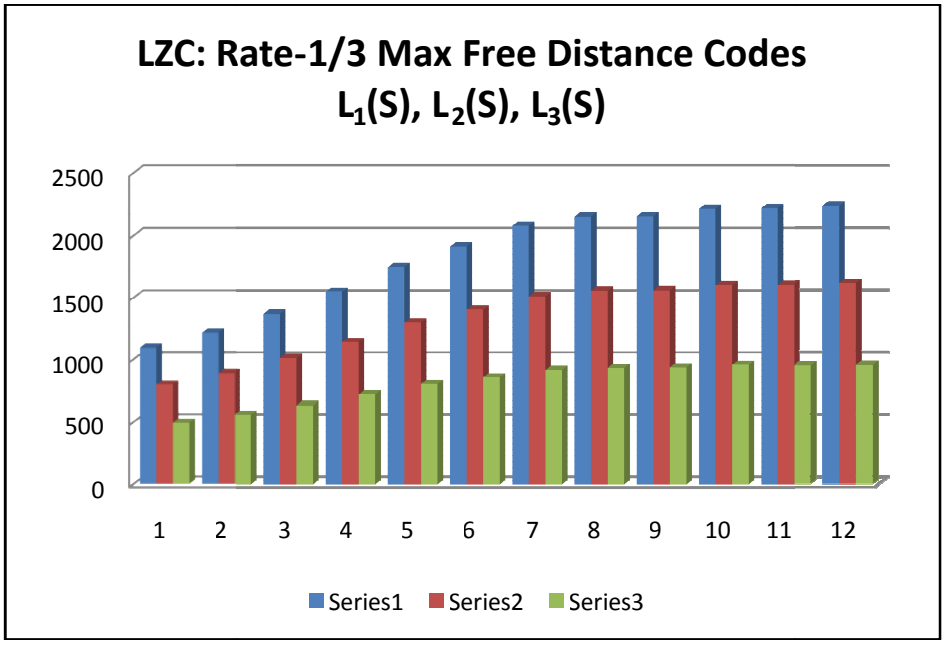


Figure 9.26 Best rate-1/3 LZ complexity for data lengths $L_1(S)$, $L_2(S)$, $L_3(S)$.

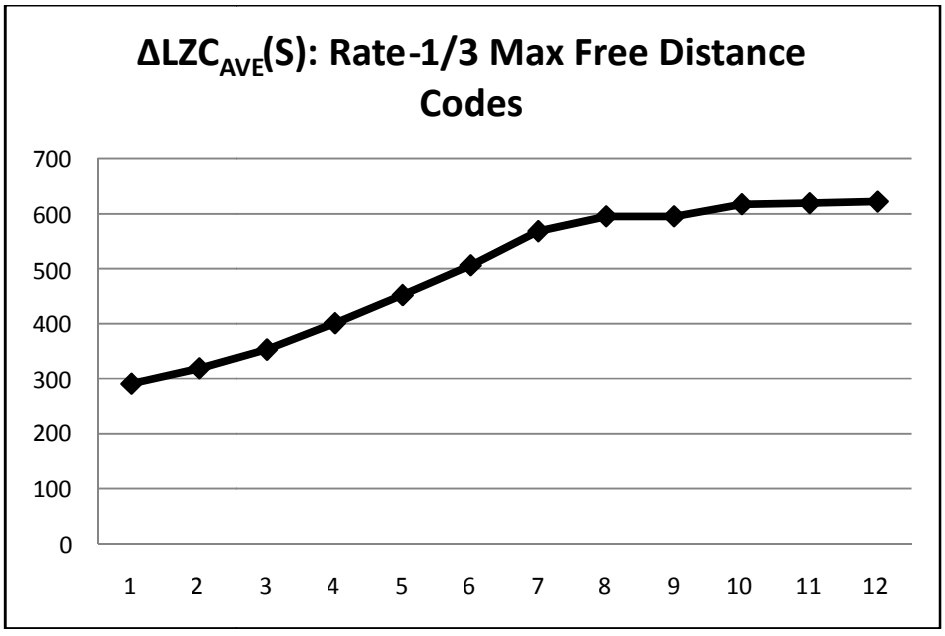


Figure 9.27 $\Delta LZC_{AVE}(S)$ for rate-1/3 maximum free distance codes.

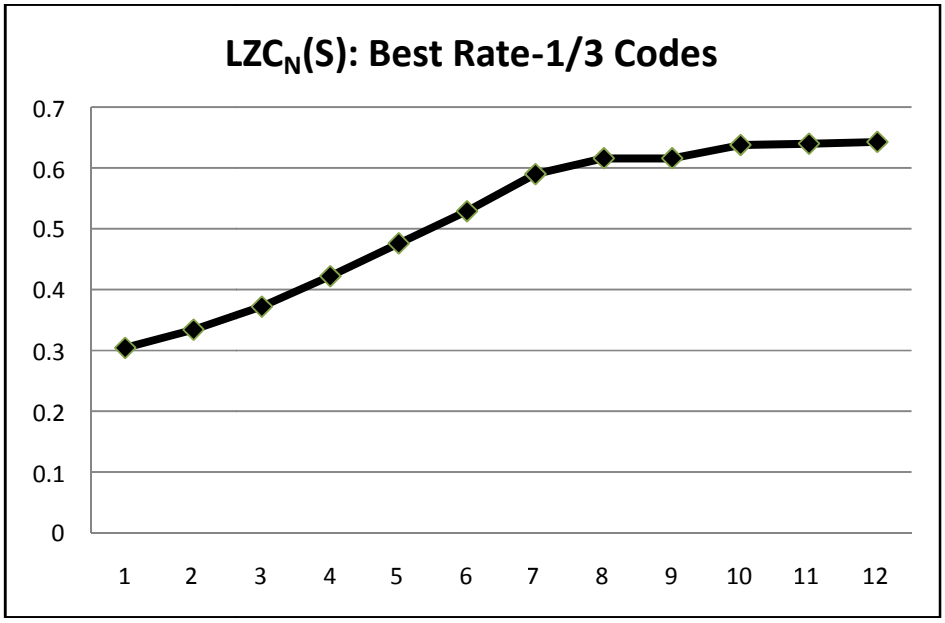


Figure 9.28 LZC_N for best 1/3-Rate codes.

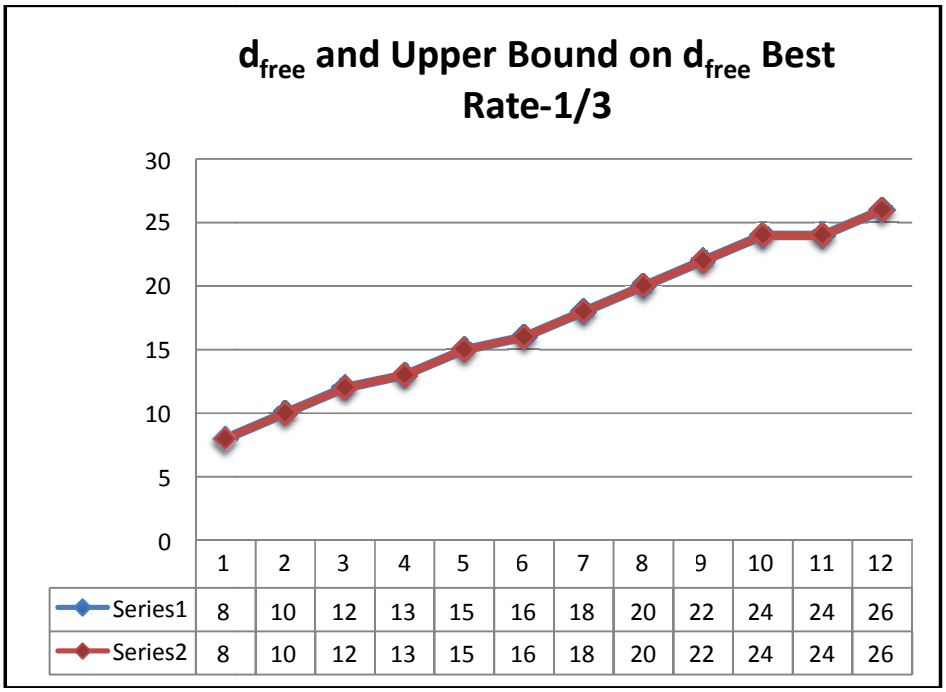


Figure 9.29 d_{free} and upper bound on d_{free} rate-1/3 max free distance codes.

Figure 9.25 shows the soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel. The complexity analysis results are given in Tables 9.8 and 9.9, in which we compare 12 different encoders across the three trials with various sequence length (Figure 9.26). This table reinforces the result that the basic path schemes provide excellent complexity convergence on average. This is mirrored in ΔLZC_{AVE} variations for the 1/3-Rate maximum free distance class codes which are depicted in Figure 9.27. Furthermore, LZC_N, d_{free} and the upper bound on d_{free} for this class are shown in Figures 9.28 and 9.29. It is interesting that the LZ complexity increment per 10000 bits, ΔLZC_{AVE}, and the normalized LZ complexity, LZC_N, both have same behavior when codes vary from $C_1^{(1/3)}$ to $C_{12}^{(1/3)}$. The least LZ complexity performance and the smallest free distance among the rate-1/3 codes belongs to $C_1^{(1/3)}$. On the other hand, the code $C_{12}^{(1/3)}$ has the highest LZ complexity and free distance value in the best rate-1/3 class codes. Codes $C_{10}^{(1/3)}$ and $C_{11}^{(1/3)}$ free distances are equal and the difference between the coresponding LZ complexites is small.

Best Rate-1/4 Class

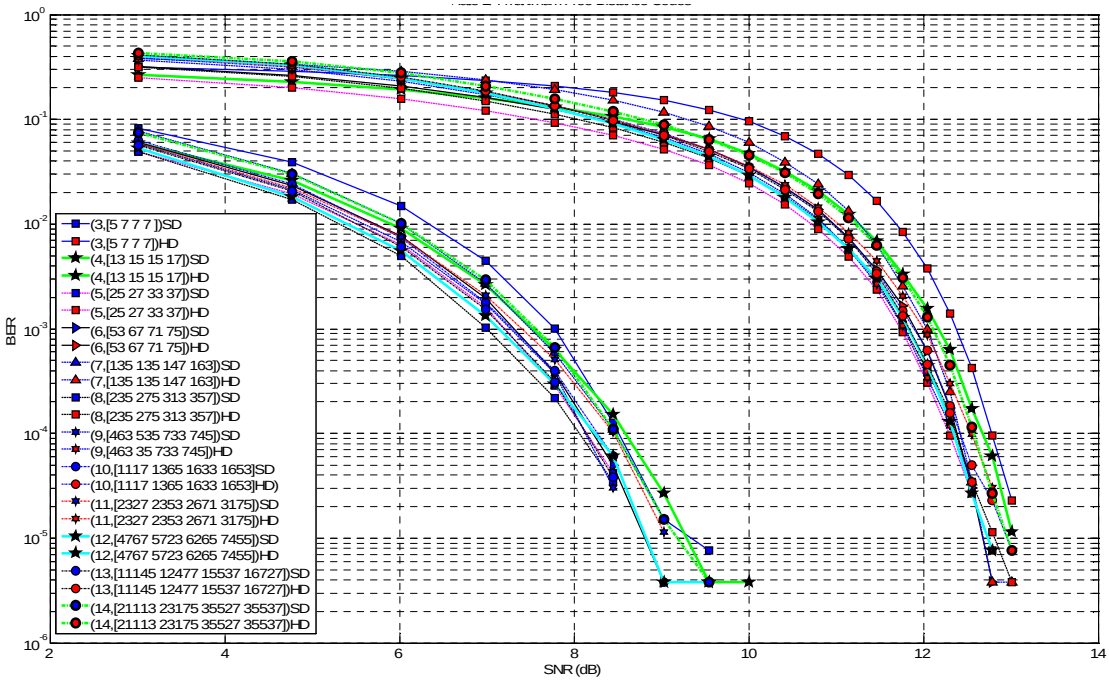


Figure 9.30 Soft decision and hard decision Viterbi decoder bit error rate performance on AWGN channel with Logistic chaos encryption (μ=4) for best rate-1/4 convolutional codes: $C_1^{(1/4)}$(3,[5 7 7 7]); $C_2^{(1/4)}$(4,[13 15 15 17]); $C_3^{(1/4)}$(5,[25 27 33 37]); $C_4^{(1/4)}$(6,[53 67 71 75]); $C_5^{(1/4)}$(7,[135 135 147 163]); $C_6^{(1/4)}$(8,[235 275 313 357]); $C_7^{(1/4)}$(9, [463 535 733 745]); $C_8^{(1/4)}$(10,[1117 1365 1633 1653]); $C_9^{(1/4)}$(11,[2327 2353 2671 3175]); $C_{10}^{(1/4)}$(12,[4767 5723 6265 7455]); $C_{11}^{(1/4)}$(13,[11145 12477 15537 16727]); $C_{12}^{(1/4)}$(14,[21113 23175 35527 35537]).

Table 9.10 Best rate-1/4 convolutional codes and their complexity measures and weight structure

$C_1^{(1/4)}$(3,[5 7 7 7]); $C_2^{(1/4)}$(4,[13 15 15 17]); $C_3^{(1/4)}$(5,[25 27 33 37]); $C_4^{(1/4)}$(6,[53 67 71 75]); $C_5^{(1/4)}$(7,[135 135 147 163]); $C_6^{(1/4)}$(8,[235 275 313 357]); $C_7^{(1/4)}$(9, [463 535 733 745]); $C_8^{(1/4)}$(10,[1117 1365 1633 1653]); $C_9^{(1/4)}$(11,[2327 2353 2671 3175]); $C_{10}^{(1/4)}$(12,[4767 5723 6265 7455]); $C_{11}^{(1/4)}$(13,[11145 12477 15537 16727]); $C_{12}^{(1/4)}$(14,[21113 23175 35527 35537])												
Rate-1/4 Class	**$C_1^{(1/4)}$**	**$C_2^{(1/4)}$**	**$C_3^{(1/4)}$**	**$C_4^{(1/4)}$**	**$C_5^{(1/4)}$**	**$C_6^{(1/4)}$**	**$C_7^{(1/4)}$**	**$C_8^{(1/4)}$**	**$C_9^{(1/4)}$**	**$C_{10}^{(1/4)}$**	**$C_{11}^{(1/4)}$**	**$C_{12}^{(1/4)}$**
$LZC\|_{L1(S)}$	862	961	1095	1255	1448	1688	1877	2037	2119	2195	2215	2230
$LZC\|_{L2(S)}$	635	711	817	940	1080	1269	1390	1490	1538	1587	1599	1607
$LZC\|_{L3(S)}$	388	445	513	605	683	790	860	908	920	950	954	957
d_{free}	10	13	16	18	20	22	24	27	29	32	33	36
Δ_{AVE} LZC per 10^4	227	250	315	368	368	419	487	547	581	608	616	623
Lim $LZC_N(S)$ S→large	0.238	0.263	0.323	0.376	0.389	0.445	0.512	0.570	0.603	0.629	0.637	0.644

In Figure 9.30, we present the soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel for rate-1/4 maximum free distance class codes. Table 9.10 provides comparison of these codes based on length, LZ complexity, LZ complexity increment per 10000 bits, normalized LZ complexity, and the codes free distance. Based on the information in this table we notice that code $C_{12}^{(1/4)}$ has achieved the best LZ complexity estimation performance in this class. The next best performing scheme, encoder $C_{11}^{(1/4)}$, requires 8% less mean estimation error capacity and 1% less mean normalized LZ complexity than the respective code $C_{12}^{(1/4)}$. Additionally, $C_{11}^{(1/4)}$ has a better time complexity which requires half the computation time needed for decoding $C_{12}^{(1/4)}$. The success of the $C_{12}^{(1/4)}$ scheme is restated in Figure 9.31. The LZ complexity increment per 10000 bits and the normalized LZ complexity are shown in Figures 9.32 and 9.33. Finally, observe that, as expected, the $C_1^{(1/4)}$ scheme maintains the lowest LZ complexity among all codes in the rate-1/4 class and worst error correcting capacity as shown in Figure 9.34.

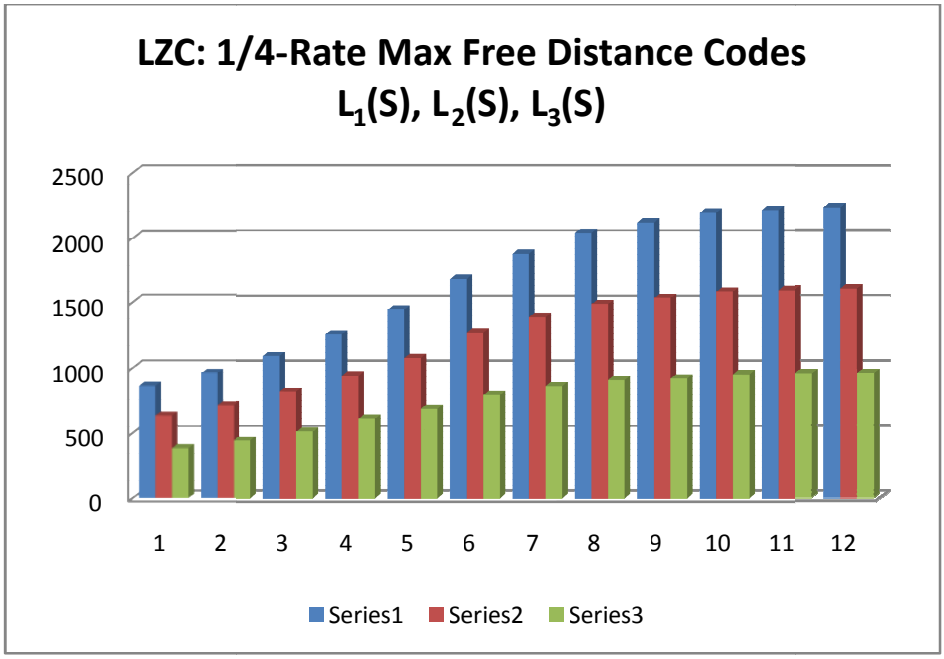


Figure 9.31 LZ complexity for the best rate-1/4 class codes.

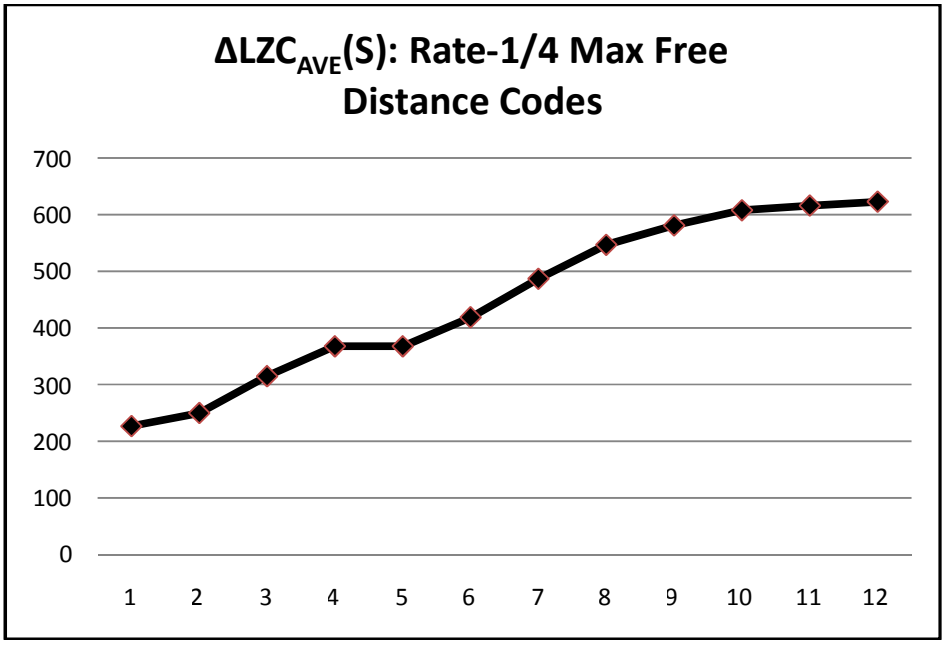


Figure 9.32 ΔLZC_{AVE} for rate-1/4 maximum free distance codes.

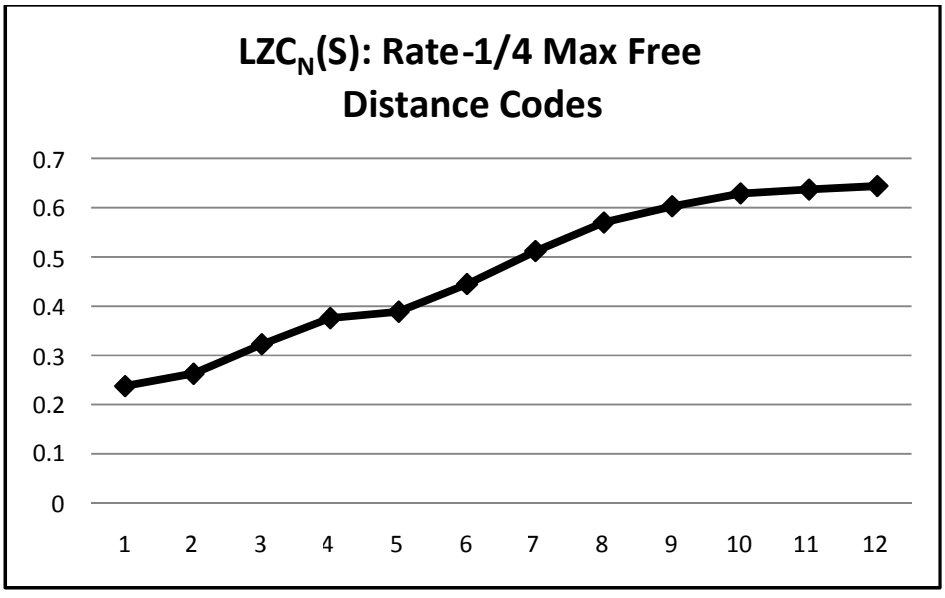


Figure 9.33 LZC_N for the best rate-1/4 class codes.

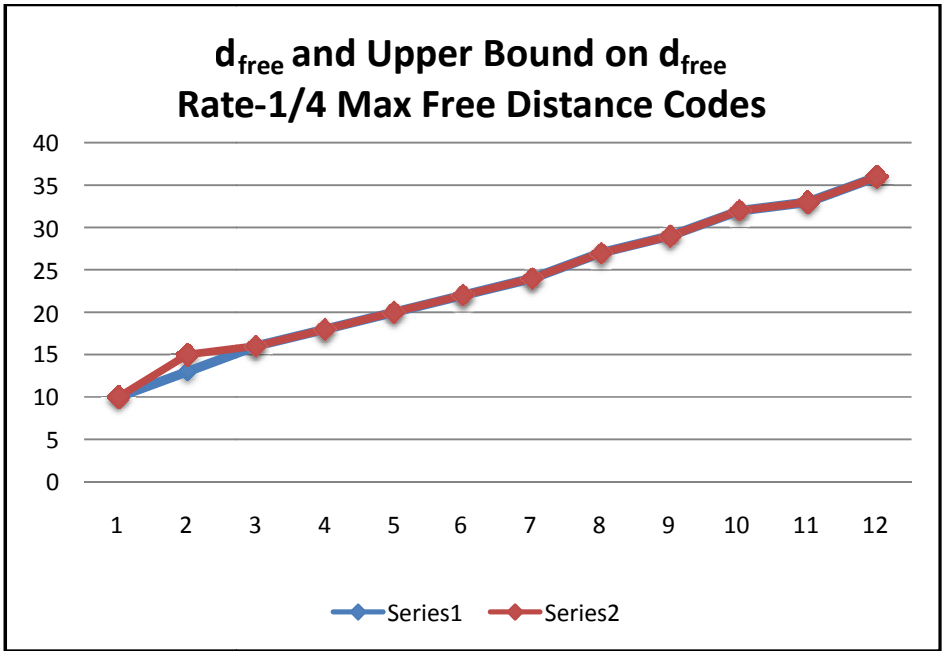


Figure 9.34 d_{free} and upper bound on d_{free} rate-1/4 max free distance codes.

Best Rate-1/5 Class

For rate-1/5 maximum free distance codes simulations, 3 trials are performed. The results are summarized in Table 9.11. Example trials for three different lengths with six encoders are shown in Figure 9.35. Also, LZ complexity increment per 10000 bits and normalized LZ complexity are shown in Figure 9.36 and 9.37. First trail with length $L_1(S)$ simply goes directly to the destination ignoring the short length effects on LZ computation. Second trial has slightly deviated from first trail result. The lowest performance encountered in third trial when a shorter sequence is allowed. But as usual it provides information about the average LZC increase rate per 10^4 bits. As expected, code $C_6^{(1/5)}$ is the best performer in providing the highest LZC as well as the largest free distance in this class. On the other hand the $C_1^{(1/5)}$ scheme maintains the lowest LZ complexity among all codes in the rate-1/5 class with smallest error correcting capacity as shown in Figure 9.38. The next best performing scheme, encoder $C_5^{(1/5)}$ requires 11% less mean estimation error capacity and 16% less mean normalized LZ complexity than the respective high

performance code $C_6^{(1/5)}$. Code $C_5^{(1/5)}$ has a better time complexity which requires half the computation time needed for decoding $C_6^{(1/5)}$. Turning back to Table 9.11, one notices that the $C_1^{(1/5)}$ scheme requires 54% less mean estimation error capacity and 50% less mean normalized LZ complexity than the respective high performance code $C_6^{(1/5)}$, but in the meantime $C_1^{(1/5)}$ has 32 times better time complexity.

Table 9.11 Best rate-1/5 convolutional codes and their complexity measures and weight structure

$C_1^{(1/5)}$(3,[7 7 7 5 5]); $C_2^{(1/5)}$(4,[17 17 13 15 15]); $C_3^{(1/5)}$(5,[37 27 33 25 35]); $C_4^{(1/5)}$(6,[75 71 73 65 57]); $C_5^{(1/5)}$(7,[175 131 135 135 147]); $C_6^{(1/5)}$(8,[257 233 323 271 357])						
Best Rate-1/5	$C_1^{(1/5)}$	$C_2^{(1/5)}$	$C_3^{(1/5)}$	$C_4^{(1/5)}$	$C_5^{(1/5)}$	$C_6^{(1/5)}$
$LZC\|_{L1(S)}$	715	798	925	1068	1261	1495
$LZC\|_{L2(S)}$	525	595	693	807	954	1133
$LZC\|_{L3(S)}$	324	373	445	526	624	730
d_{free}	13	16	20	22	25	28
Δ_{AVE} LZC per 10^4	195	212	240	271	318	382
Lim $LZC_N(S)$ S→large	0.202	0.222	0.252	0.287	0.337	0.403

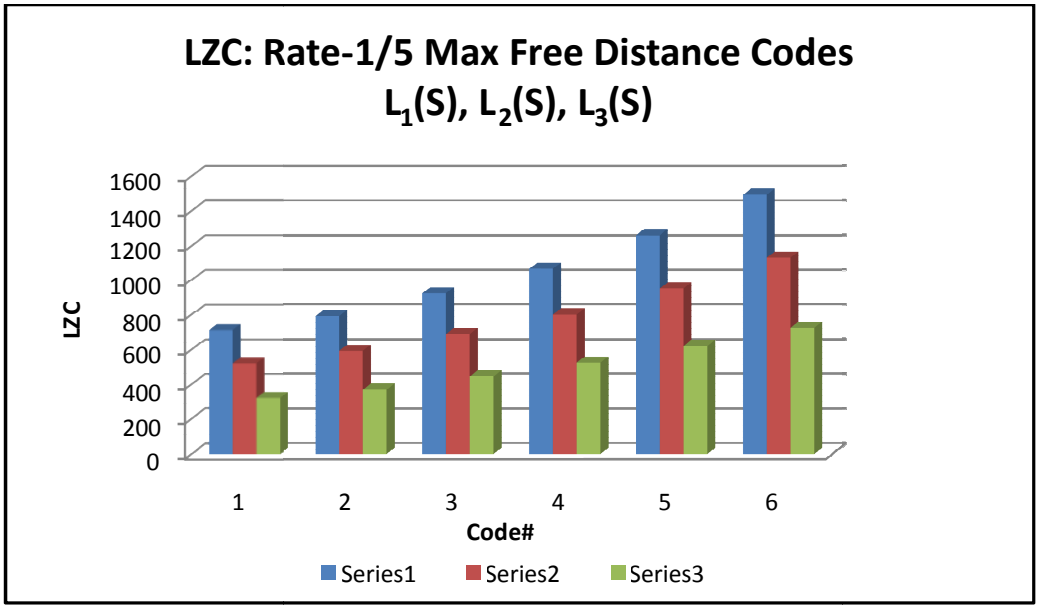


Figure 9.35 LZ complexity for best rate-1/5 class.

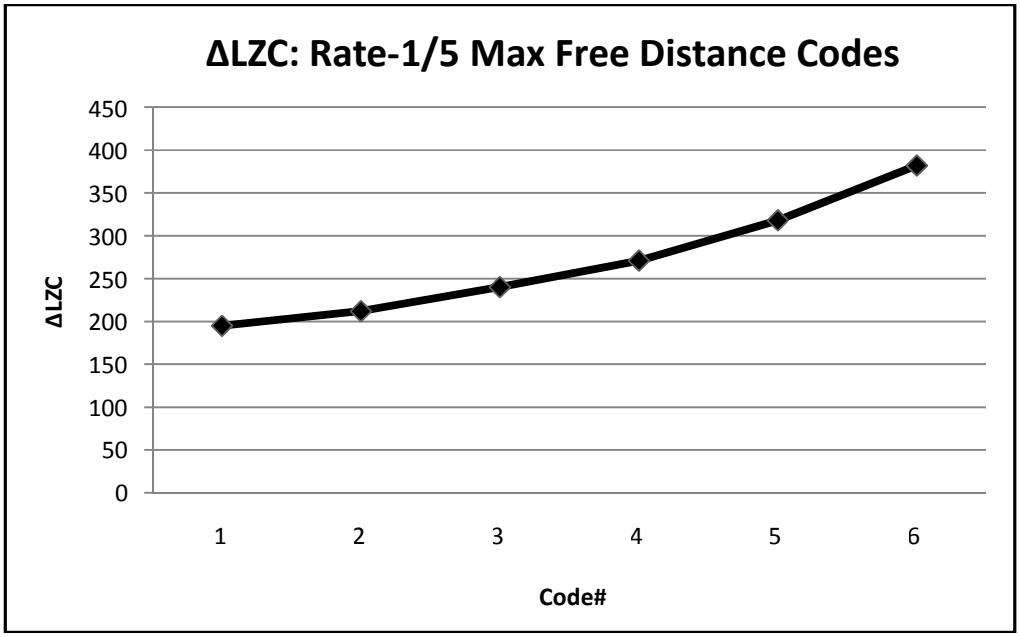


Figure 9.36 ΔLZC_{AVE} for rate-1/5 maximum free distance codes

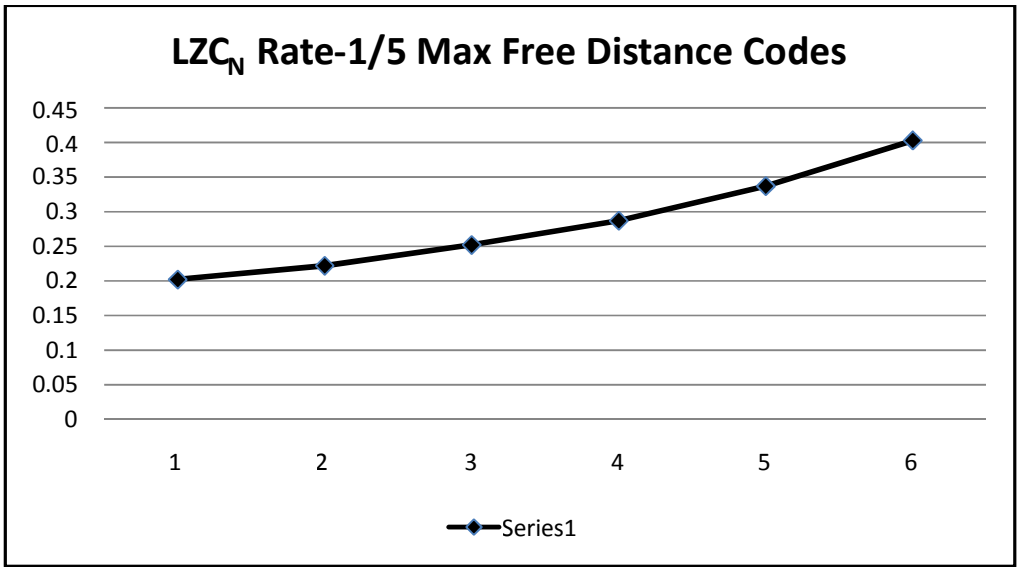


Figure 9.37 LZC_N for the best rate-1/5 class.

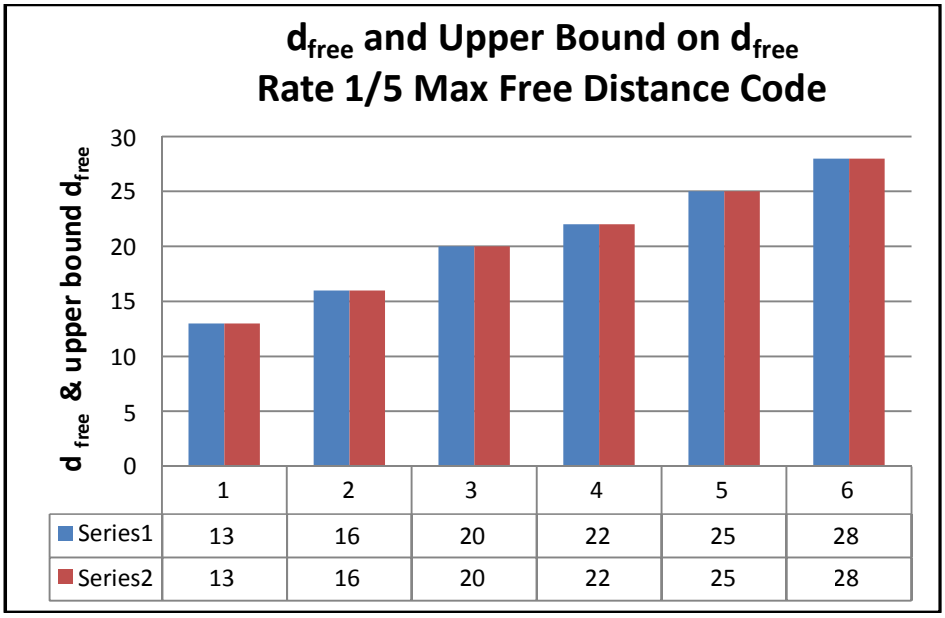


Figure 9.38 d_{free} and upper bound on d_{free} best rate-1/5 class.

Best Rate-1/6 Class

For the rate-1/6 maximum free distance codes the simulation results of three trials with different sequence lengths on six different codes are summarized in Table 9.12. It is observed from Figure 9.39 that in all the trails, $C_1^{(1/6)}$ code achieves the lowest position in estimation error in the best 1/6-rate class while code $C_6^{(1/6)}$ has the most complex responses to the input data followed by codes $C_5^{(1/6)}$ with the highest error correcting capacity as shown in Figure 9.40. The LZ complexity increment per 10000 bits and the normalized LZ complexity are shown in Figures 9.41 and 9.42. The next best performing scheme, encoder $C_5^{(1/6)}$, provides 12% less mean estimation error capacity and 18% less mean normalized LZ complexity than the respective high performance code, $C_6^{(1/6)}$. Again code $C_5^{(1/6)}$ has a better time complexity which requires half the computation time needed for decoding $C_6^{(1/6)}$. Using Tables 9.12, we notice that code $C_1^{(1/6)}$ scheme achieves 54% less mean estimation error capacity and 53% less mean normalized LZ complexity compared to code $C_6^{(1/5)}$, but provides better time complexity.

Table 9.12 Best rate-1/6 convolutional codes and their complexity measures and weight structure

$C_1^{(1/6)}$ (3,[7 7 7 7 5 5]); $C_2^{(1/6)}$ (4,[17 17 13 13 15 15]); $C_3^{(1/6)}$(5,[37 35 27 33 25 35]); $C_4^{(1/6)}$(6,[73 75 55 65 47 57]); $C_5^{(1/6)}$(7,[173 151 135 135 163 137]); $C_5^{(1/6)}$(8,[253 375 331 235 313 357])						
Best 1/6-Rate	$C_1^{(1/6)}$	$C_2^{(1/6)}$	$C_3^{(1/6)}$	$C_4^{(1/6)}$	$C_5^{(1/6)}$	$C_6^{(1/6)}$
$LZC\|_{L1(S)}$	615	678	809	942	1121	1331
$LZC\|_{L2(S)}$	453	512	606	718	860	1011
$LZC\|_{L3(S)}$	282	321	395	477	570	657
d_{free}	16	20	24	27	30	34
Δ_{AVE} LZC per 10^4	166	183	207	232	275	337
Lim $LZC_N(S)$ $S \rightarrow large$	0.173	0.190	0.219	0.247	0.294	0.357

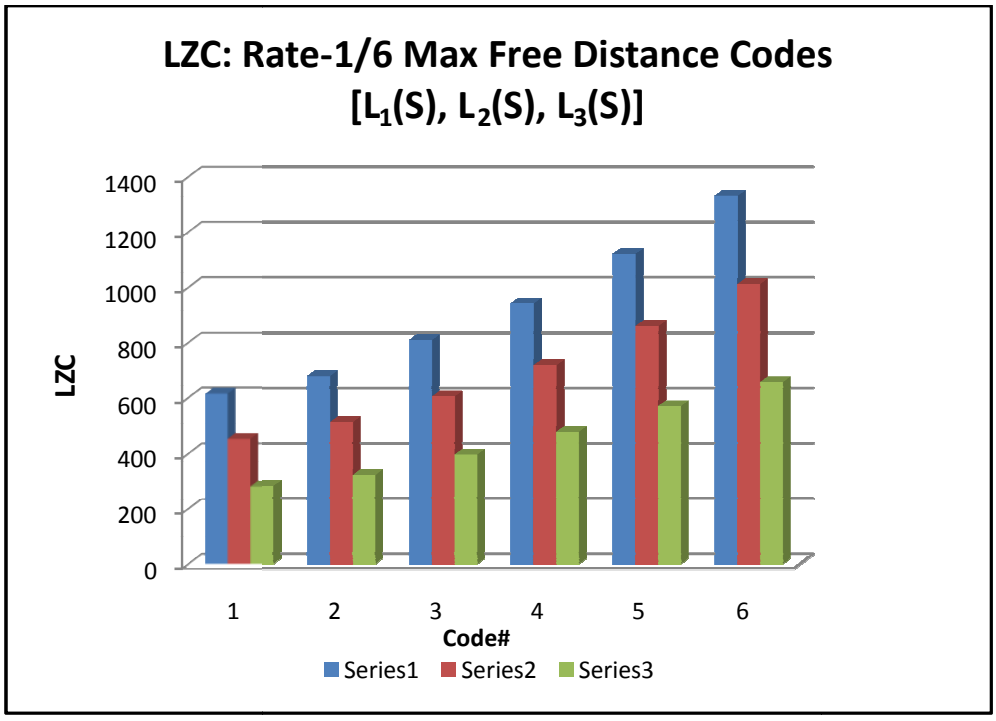


Figure 9.39 LZ complexity for best rate-1/6 class codes.

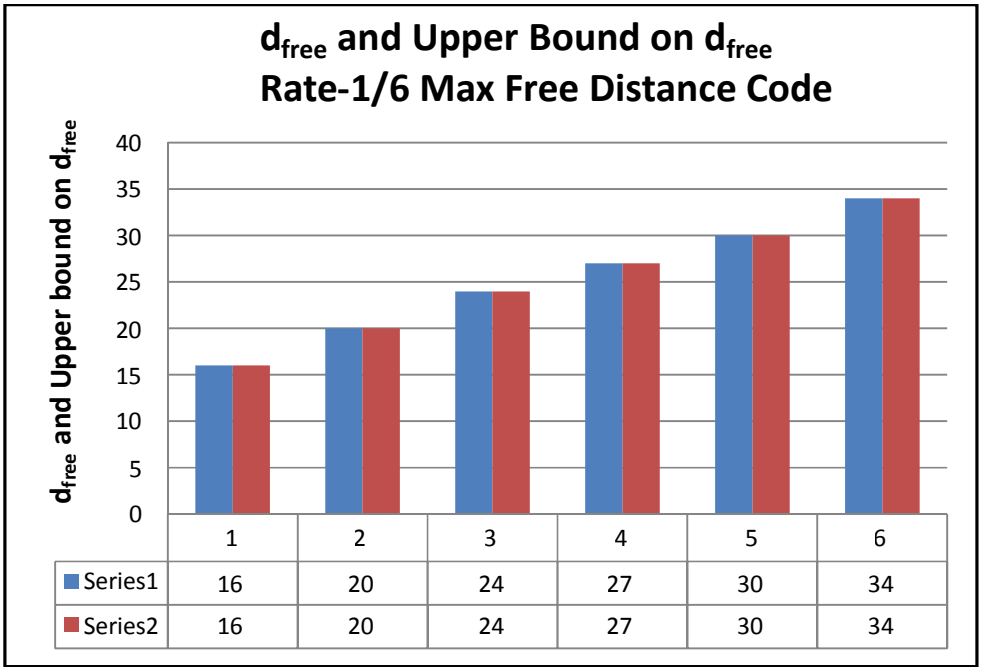


Figure 9.40 d_{free} and upper bound on d_{free} best rate-1/6 class.

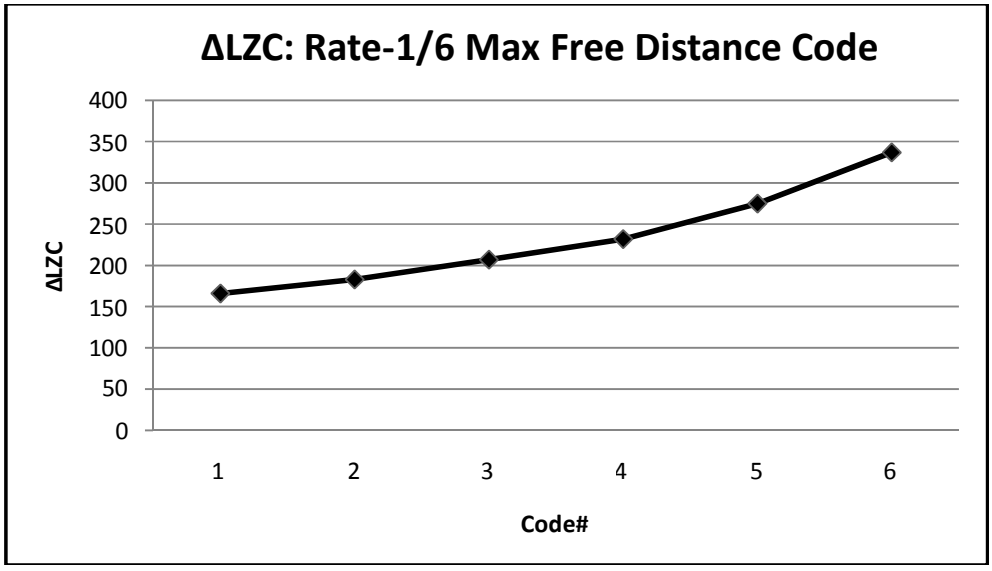


Figure 9.41 ΔLZC_{AVE} for rate-1/6 maximum free distance codes.

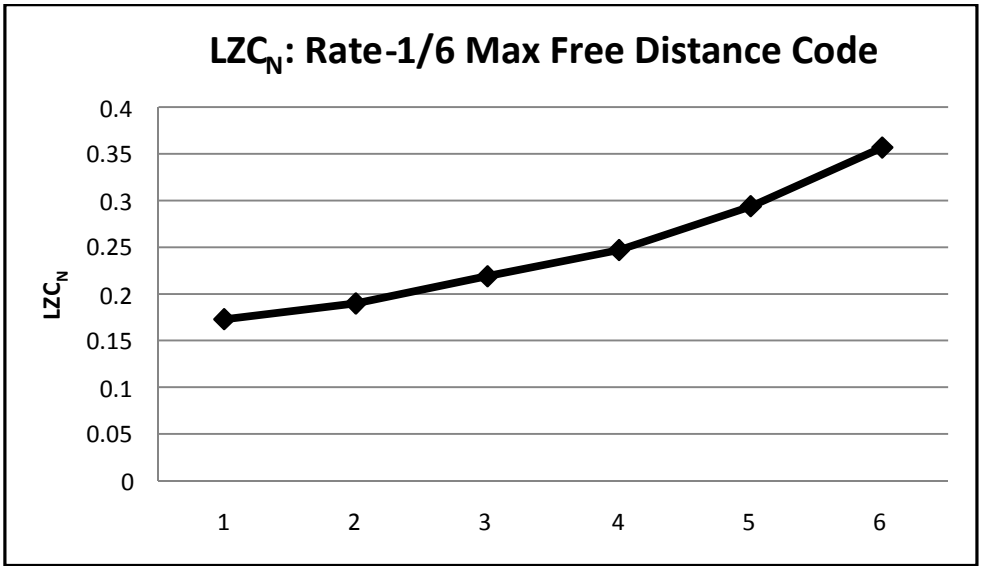


Figure 9.42 LZC_N for the best rate-1/6 class.

Best Rate-1/7 Class

The results of these simulation trials on three distinct sequence lengths are listed in Table 9.13. As expected, code $C_1^{(1/7)}$ is the quickest scheme and $C_6^{(1/7)}$ is the slowest one to decode the destination sequence.

Table 9.13 Best rate-1/7 convolutional codes and their complexity measures and weight structure.

$C_1^{(1/7)}$(3,[5 5 5 7 7 7 7]); $C_2^{(1/7)}$ (4,[13 13 13 15 15 17 17]); $C_3^{(1/7)}$(5,[35 27 25 27 33 35 37]); $C_4^{(1/7)}$(6,[73 75 55 65 47 57]); $C_5^{(1/7)}$(7,[165 145 173 135 147 137]); $C_6^{(1/7)}$(8,[275 253 375 235 313 357])						
Best 1/7-Rate	**$C_1^{(1/7)}$**	**$C_2^{(1/7)}$**	**$C_3^{(1/7)}$**	**$C_4^{(1/7)}$**	**$C_5^{(1/7)}$**	**$C_6^{(1/7)}$**
$LZC\|_{L1(S)}$	536	612	720	942	1119	1352
$LZC\|_{L2(S)}$	395	457	543	718	858	1030
$LZC\|_{L3(S)}$	250	291	363	477	567	675
d_{free}	18	23	28	32	36	40
Δ_{AVE} LZC per 10^4	143	160	178	232	276	338
Lim $LZC_N(S)$ $S \rightarrow$large	0.149	0.168	0.189	0.247	0.343	0.359

Once the desired decoded data are reached the $C_6^{(1/7)}$ encoder demonstrates the highest LZ complexity and error corecting performance in this class as shown in Figures 9.43 and 9.44. The LZ complexity increment per 10000 bits and the normalized LZ complexity are shown in Figures 9.45 and 9.46. Encoder $C_5^{(1/7)}$ has the next best performing scheme, where it provides 10% less mean estimation error capacity and only 5% less mean normalized LZ complexity than the respective high performance code $C_6^{(1/7)}$. Once again $C_5^{(1/7)}$ performs better in time complexity which requires half the computation time needed for decoding $C_6^{(1/7)}$. From the data collected in Table 9.13, it is seen that $C_1^{(1/5)}$ requires 55% less mean estimation error capacity and 59% less mean normalized LZ complexity than the respective high performance code $C_6^{(1/7)}$. In Figure 9.47, we present the soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel for 1/5, 1/6, 1/7-rate maximum free distance class codes.

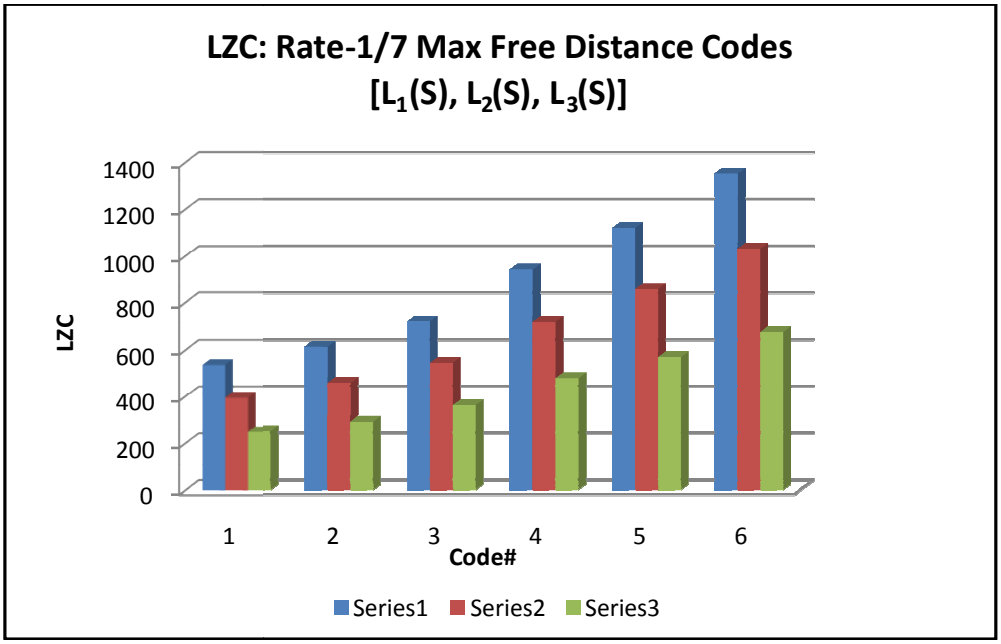


Figure 9.43 LZC for best rate-1/7 class codes.

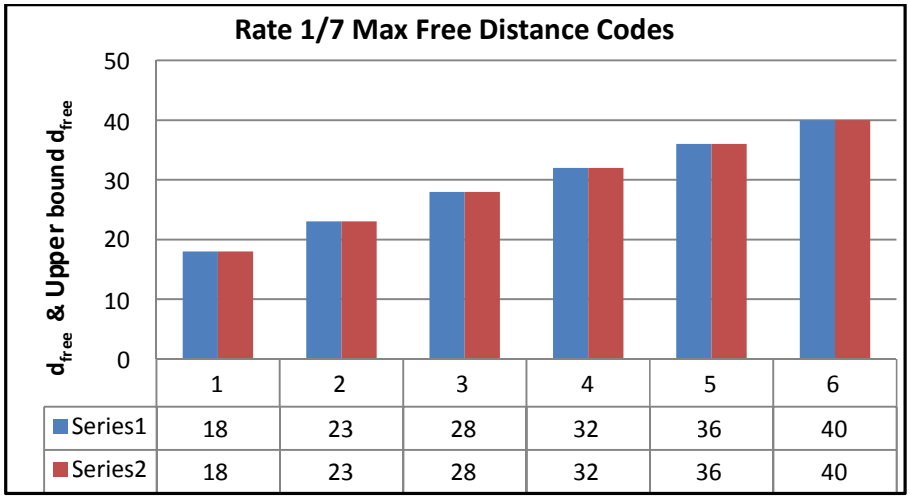


Figure 9.44 d_{free} and upper bound on d_{free} best rate-1/7 class.

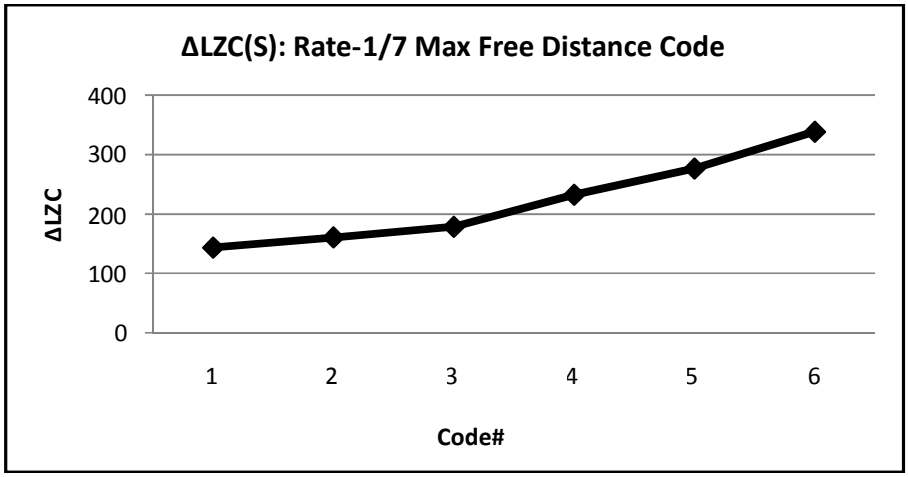


Figure 9.45 ΔLZC_{AVE} for rate-1/7 maximum free distance codes.

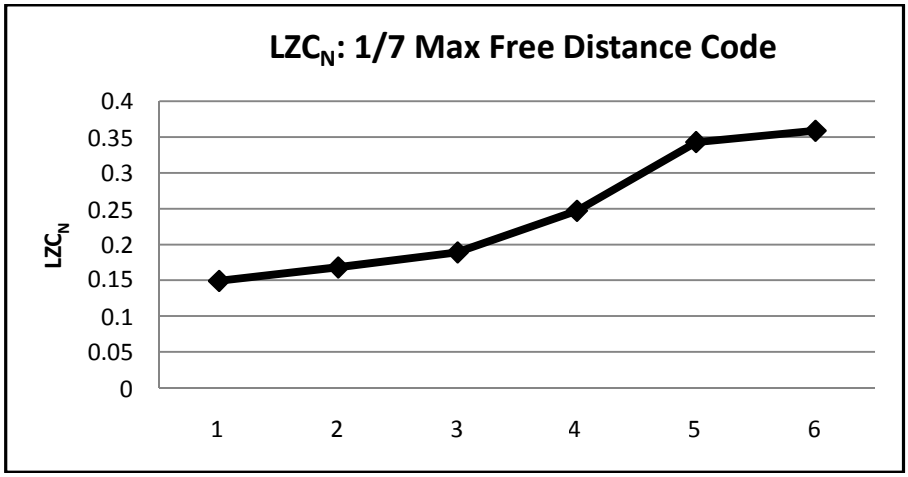


Figure 9.46 LZC_N for the best rate-1/7 class.

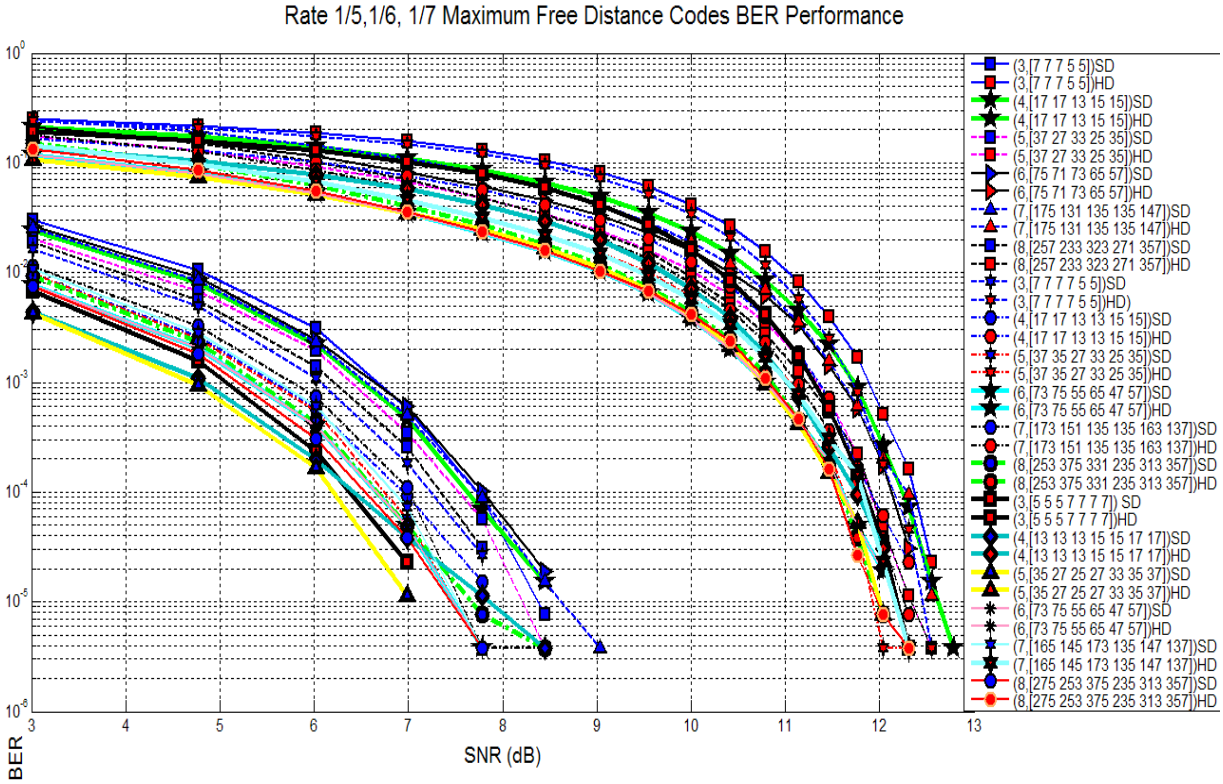


Figure 9.47 Soft decision and hard decision Viterbi decoder bit error rate performance on an AWGN channel with Logistic chaos encryption (μ=4) for the best rate-1/5, 1/6, 1/7 convolutional codes. Best rate-1/5 class codes: $C_1^{(1/5)}$(3,[7 7 7 5 5]), $C_2^{(1/5)}$(4,[17 17 13 15 15]), $C_3^{(1/5)}$(5,[37 27 33 25 35]), $C_4^{(1/5)}$(6,[75 71 73 65 57]), $C_5^{(1/5)}$(7,[175 131 135 135 147]), $C_6^{(1/5)}$(8,[257 233 323 271 357]). Best rate-1/6 class codes: $C_1^{(1/6)}$ (3,[7 7 7 7 5 5]), $C_2^{(1/6)}$ (4,[17 17 13 13 15 15]), $C_3^{(1/6)}$(5,[37 35 27 33 25 35]), $C_4^{(1/6)}$(6,[73 75 55 65 47 57]), $C_5^{(1/6)}$(7,[173 151 135 135 163 137]), $C_6^{(1/6)}$(8,[253 375 331 235 313 357]). Best rate-1/7 class: $C_1^{(1/7)}$(3,[5 5 5 7 7 7 7]), $C_2^{(1/7)}$ (4,[13 13 13 15 15 17 17]), $C_3^{(1/7)}$(5,[35 27 25 27 33 35 37]), $C_4^{(1/7)}$(6,[73 75 55 65 47 57]), $C_5^{(1/7)}$(7,[165 145 173 135 147 137]), $C_6^{(1/7)}$(8,[275 253 375 235 313 357]).

9. 15 Summary of Results in Complexity Analysis of the Best Rate-1/2, 1/3, 1/4, 1/5, 1/6, 1/7

In this section we summarize the simulation results obtained in previous sections to provide a complete picture of the overall performance of best rate-1/*n* codes. In previous sections in this chapter special attention was paid to the algorithmic complexity of an information sequence for an entire class of rate-1/*n* encoders. Additionally, we have investigated the computational complexity of each decoding algorithm and compared the results to an alternative

algorithm. The following argument provides an explanation for the computational complexity increase as constraint length increases. Consider a binary convolutional code with one input stream line ($k = 1$) and constraint length of K is decoded by means of the Viterbi algorithm, there will be $2^{(K-1)}$ states. Hence, there are $2^{(K-1)}$ surviving paths at each stage and $2^{(K-1)}$ metrics, one for each surviving path. Furthermore, a binary convolutional code in which one bit (k=1) at a time is shifted into an encoder that consists of K (1-bit) shift-register stages generates a trellis that has $2^{(K-1)}$ states. Consequently, the decoding of such a code by means of the Viterbi algorithm requires keeping track of $2^{(K-1)}$ surviving paths and $2^{(K-1)}$ metrics. At each stage of the trellis, there are two paths that merge at each node. Since each path that converges at a common node requires the computation of a metric, there are two metrics computed for each node. Out of the two paths that merge at each node, only one survives, and this is the most-probable (minimum-distance) path. Thus, the number of computations in decoding performed at each stage increases exponentially with constraint length (K). Our numerical evidence indicates clearly algorithmic complexity associated with particular rate-1/n convolutional encoders increases as constraint length increases (Figures 9.47 and 9.48), while error correcting capacity of the decoder expands (Figure 9.49). This proves that algorithmic complexity is a suitable measure of the quality factor along with other measures of the weight structure of the code. For example, when maximum likelihood decoding is used, the optimum weight structure is that which has the minimum number of bit errors in the paths through the code trellis which are closest to one another in Hamming distance and the free distance is the minimum Hamming distance between the code words on any two paths through the trellis. Concerning a group of various rate-1/n convolutional encoders, we find that as the number of output streams, n increases from 2 (or 1/n decreases from 1/2 to 1/7 rate codes) free distance also increases while the encoded data sequence manifests low

complexity in its algorithm. Turning back to Figures 9.47, 9.48 and 9.50 notice that the codes $C_{12}^{(1/2)}$, $C_{12}^{(1/3)}$, $C_{12}^{(1/4)}$ explicitly have achieved the best algorithmic complexity performance. In particular, these schemes have at most 22% difference in this LZC factor but their difference in error correcting capacity is significant. For example, $C_{12}^{(1/2)}$ has 39% less error correcting capacity than $C_{12}^{(1/3)}$, $C_{12}^{(1/2)}$ has 55% less than $C_{12}^{(1/4)}$ and $C_{12}^{(1/3)}$ has 28% less than $C_{12}^{(1/4)}$. Similar observation in the performance of $C_4^{(1/6)}$, $C_4^{(1/7)}$, $C_5^{(1/6)}$ $C_5^{(1/7)}$, $C_6^{(1/6)}$, $C_6^{(1/7)}$ indicates that despite the significant diffrence in error corecting performance, the algorithmic complexity of the decoder output sequence has almost same performance. Additionally, it is encouraging that the class rate-1/5 codes only require a modest number of operations, in particular consider $C_6^{(1/5)}$ with a modest LZ complexity estimation and good free distance and better time complexity achievement compared to all codes in classes $C^{(1/3)}$, $C^{(1/2)}$ and some codes in class $C^{(1/4)}$ such as $C_{j \leq 8}^{(1/5)}$ codes. The success of the best rate-1/n schemes are reinforced by Figures 9.47 through 9.50. Figure 9.51 illustrates the d_{free} for the best rate-1/*n* class codes. Finally, observe that, as expected, the class $C^{(1/7)}$ encoding scheme maintains a lower estimation in LZ complexity at most of the constraint lengths than any other scheme except some codes in class $C^{(1/6)}$ as shown in Figure 9.49. The simulations have confirmed that both code design and constraint length are control architectural parameters that have significant effect on system performance. These components of the unified performance evaluation discussed here have proved to be beneficial for a system operating in realistic simulations.

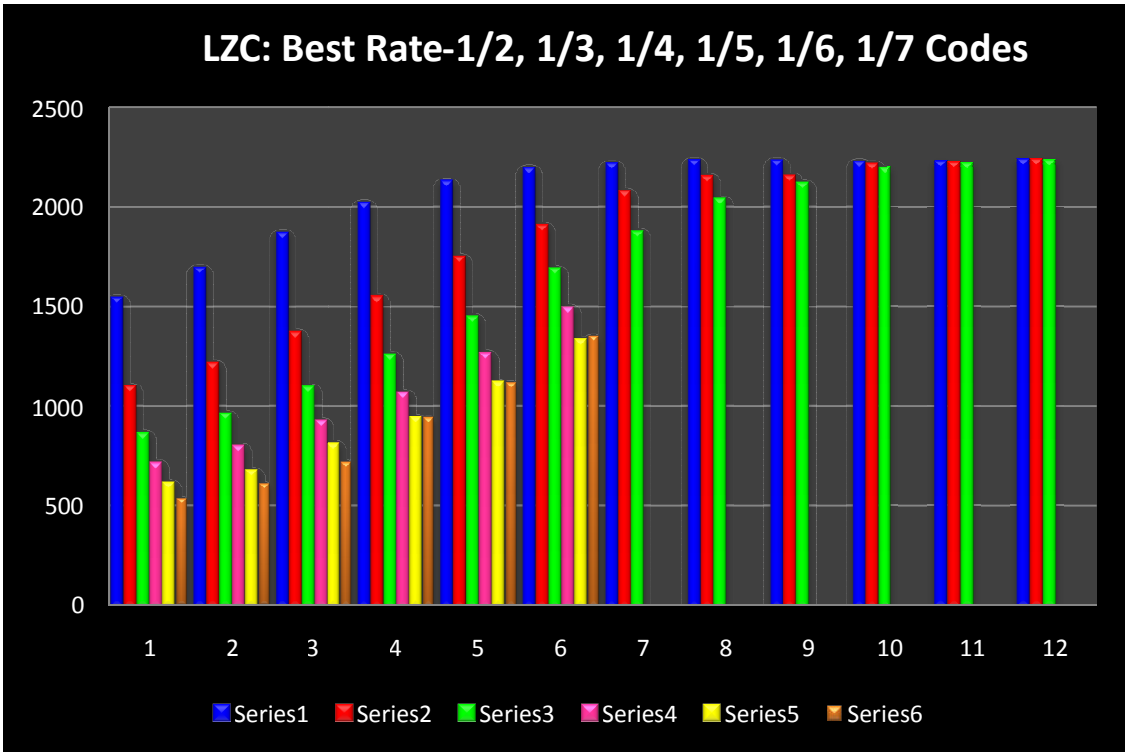
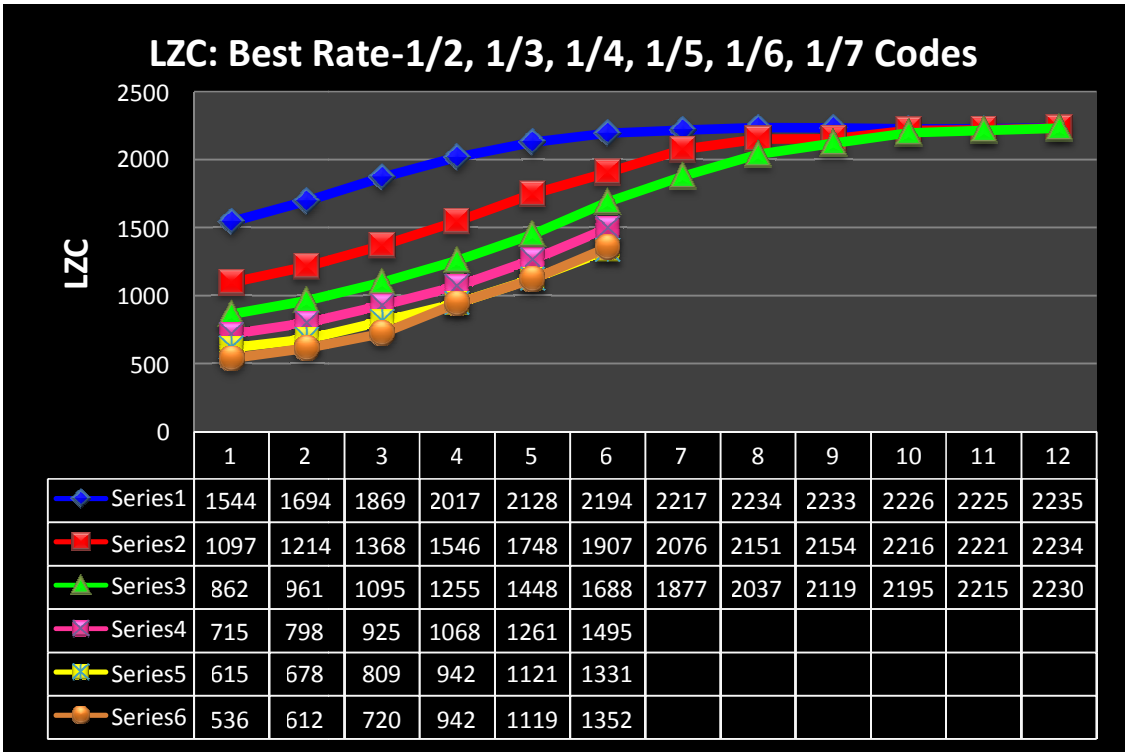
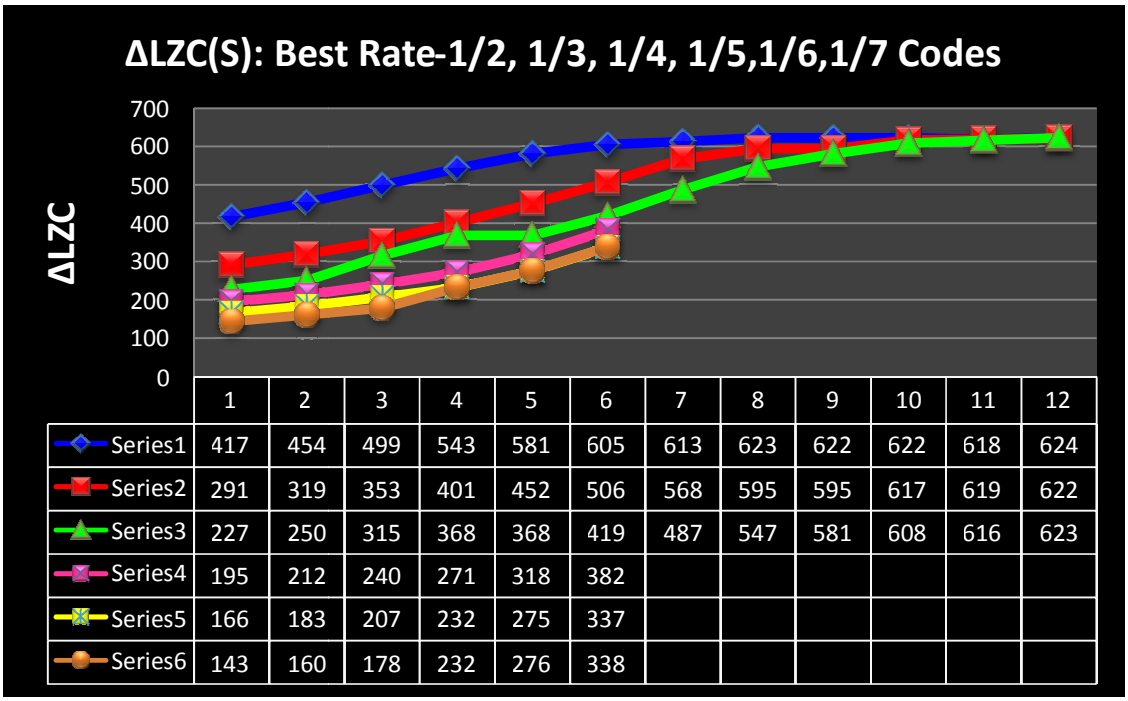


Figure 9.48 Classes of various rate-1/*n* convolutional encoders. Best rate-1/2 convolutional codes: $C_1^{(1/2)}$ (3, [5 7]); $C_2^{(1/2)}$ (4, [15 17]); $C_3^{(1/2)}$ (5, [23 35]); $C_4^{(1/2)}$ (6, [53 75]); $C_5^{(1/2)}$ (7, [133 171]); $C_6^{(1/2)}$ (8, [247 371]); $C_7^{(1/2)}$ (9, [561 753]); $C_8^{(1/2)}$ (10, ,[1167 1545]); $C_9^{(1/2)}$ (11,[2335 3661]); $C_{10}^{(1/2)}$(12,[4335 5723]); $C_{11}^{(1/2)}$(13,[10533 17661]); $C_{12}^{(1/2)}$(14,[21675 27123]). Best rate-1/3 convolutional codes: $C_1^{(1/3)}$(3,[5 7 7]); $C_2^{(1/3)}$(4,[13 15 17]); $C_3^{(1/3)}$(5,[25 33 37]); $C_4^{(1/3)}$(6,[47 53 75]); $C_5^{(1/3)}$ (7,[133 145 175]); $C_6^{(1/3)}$(8,[225 331 367]); $C_7^{(1/3)}$(9, [557 663 711]); $C_8^{(1/3)}$ (10,[1117 1365 1633]); $C_9^{(1/3)}$(11,[2353 2671 3175]); $C_{10}^{(1/3)}$(12,[4767 5723 6265]); $C_{11}^{(1/3)}$(13,[10533 10675 17661]); $C_{12}^{(1/3)}$ (14,[21645 35661 37133]). Best rate-1/4 convolutional codes: $C_1^{(1/4)}$(3,[5 7 7 7]); $C_2^{(1/4)}$(4,[13 15 15 17]); $C_3^{(1/4)}$(5,[25 27 33 37]); $C_4^{(1/4)}$(6,[53 67 71 75]); $C_5^{(1/4)}$(7,[135 135 147 163]); $C_6^{(1/4)}$(8,[235 275 313 357]); $C_7^{(1/4)}$(9, [463 535 733 745]); $C_8^{(1/4)}$(10,[1117 1365 1633 1653]); $C_9^{(1/4)}$(11,[2327 2353 2671 3175]); $C_{10}^{(1/4)}$(12,[4767 5723 6265 7455]); $C_{11}^{(1/4)}$(13,[11145 12477 15537 16727]); $C_{12}^{(1/4)}$(14,[21113 23175 35527 35537]). Best rate-1/5 class codes: $C_1^{(1/5)}$(3,[7 7 7 5 5]), $C_2^{(1/5)}$(4,[17 17 13 15 15]), $C_3^{(1/5)}$(5,[37 27 33 25 35]), $C_4^{(1/5)}$(6,[75 71 73 65 57]), $C_5^{(1/5)}$(7,[175 131 135 135 147]), $C_6^{(1/5)}$(8,[257 233 323 271 357]). Best rate-1/6 class codes: $C_1^{(1/6)}$(3,[7 7 7 7 5 5]), $C_2^{(1/6)}$(4,[17 17 13 13 15 15]), $C_3^{(1/6)}$(5,[37 35 27 33 25 35]), $C_4^{(1/6)}$(6,[73 75 55 65 47 57]), $C_5^{(1/6)}$(7,[173 151 135 135 163 137]), $C_6^{(1/6)}$(8,[253 375 331 235 313 357]). Best rate-1/7 class: $C_1^{(1/7)}$(3,[5 5 5 7 7 7 7]), $C_2^{(1/7)}$(4,[13 13 13 15 15 17 17]), $C_3^{(1/7)}$(5,[35 27 25 27 33 35 37]), $C_4^{(1/7)}$(6,[73 75 55 65 47 57]), $C_5^{(1/7)}$(7,[165 145 173 135 147 137]), $C_6^{(1/7)}$(8,[275 253 375 235 313 357]).



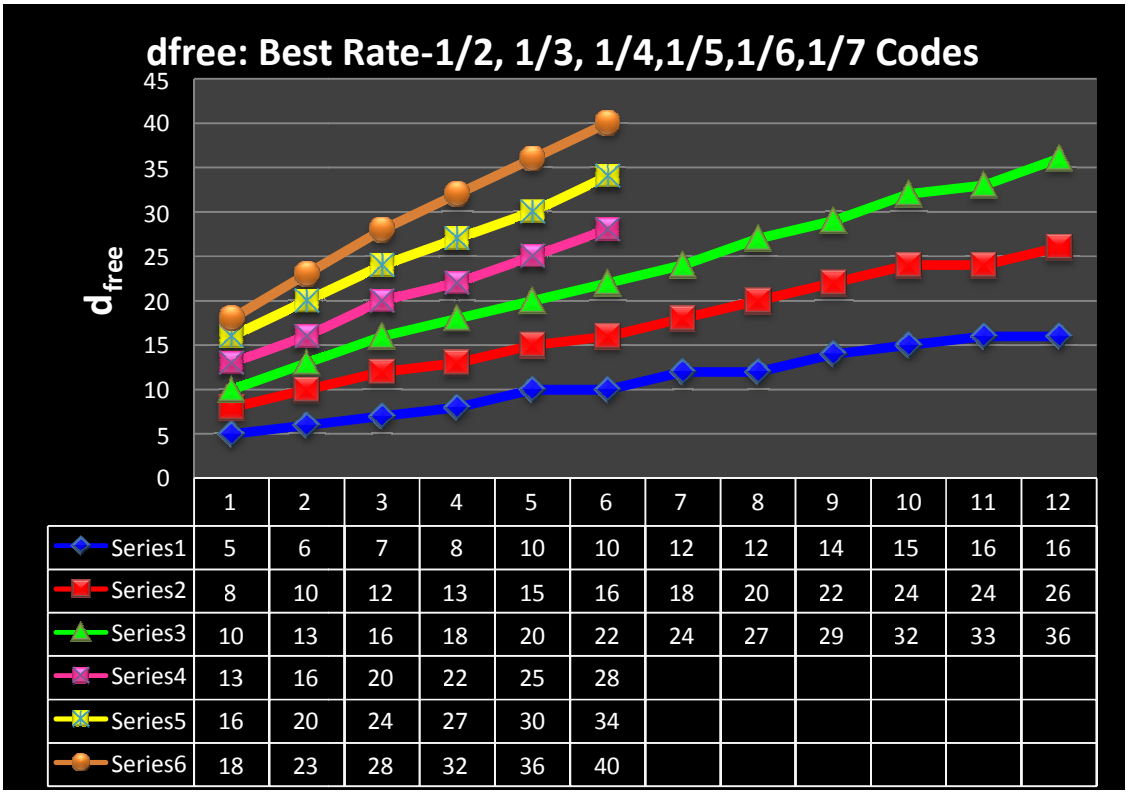
	1	2	3	4	5	6	7	8	9	10	11	12
Series1	1544	1694	1869	2017	2128	2194	2217	2234	2233	2226	2225	2235
Series2	1097	1214	1368	1546	1748	1907	2076	2151	2154	2216	2221	2234
Series3	862	961	1095	1255	1448	1688	1877	2037	2119	2195	2215	2230
Series4	715	798	925	1068	1261	1495						
Series5	615	678	809	942	1121	1331						
Series6	536	612	720	942	1119	1352						

Figure 9.49 Classes of various rate-1/*n* convolutional encoders.



	1	2	3	4	5	6	7	8	9	10	11	12
Series1	417	454	499	543	581	605	613	623	622	622	618	624
Series2	291	319	353	401	452	506	568	595	595	617	619	622
Series3	227	250	315	368	368	419	487	547	581	608	616	623
Series4	195	212	240	271	318	382						
Series5	166	183	207	232	275	337						
Series6	143	160	178	232	276	338						

Figure 9.50 $\Delta LZC_{AVE}(S)$ for rate-1/*n* maximum free distance codes.



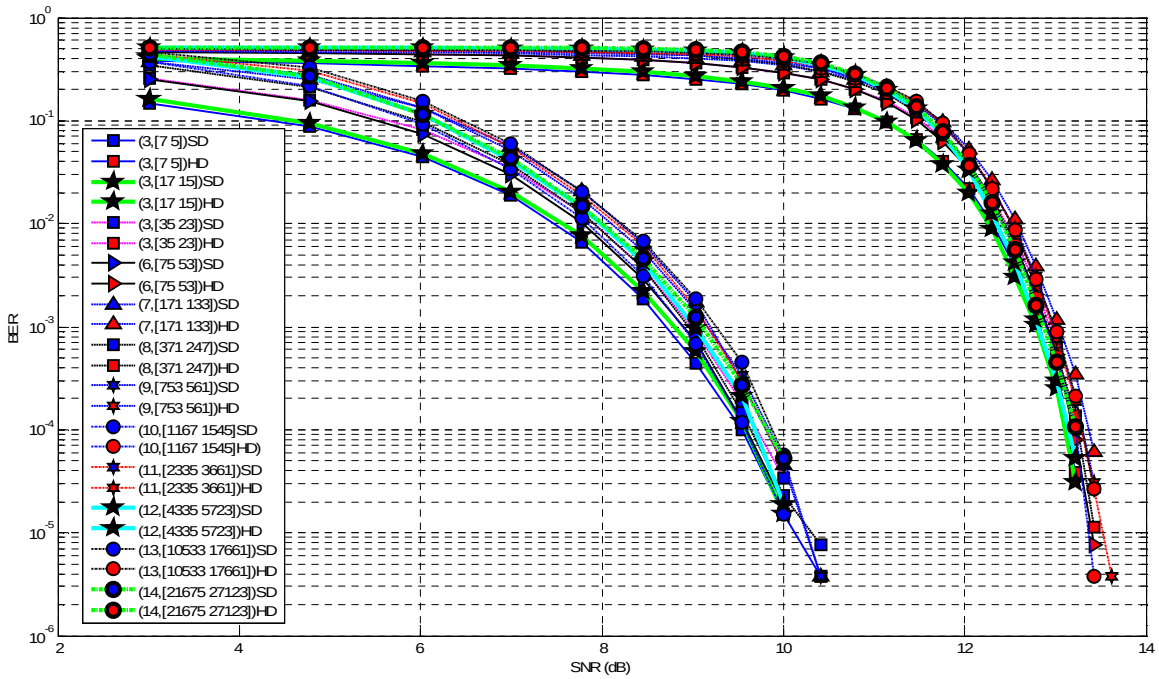
	1	2	3	4	5	6	7	8	9	10	11	12
Series1	5	6	7	8	10	10	12	12	14	15	16	16
Series2	8	10	12	13	15	16	18	20	22	24	24	26
Series3	10	13	16	18	20	22	24	27	29	32	33	36
Series4	13	16	20	22	25	28						
Series5	16	20	24	27	30	34						
Series6	18	23	28	32	36	40						

Figure 9.51 d_{free} for the best rate-1/*n* class.

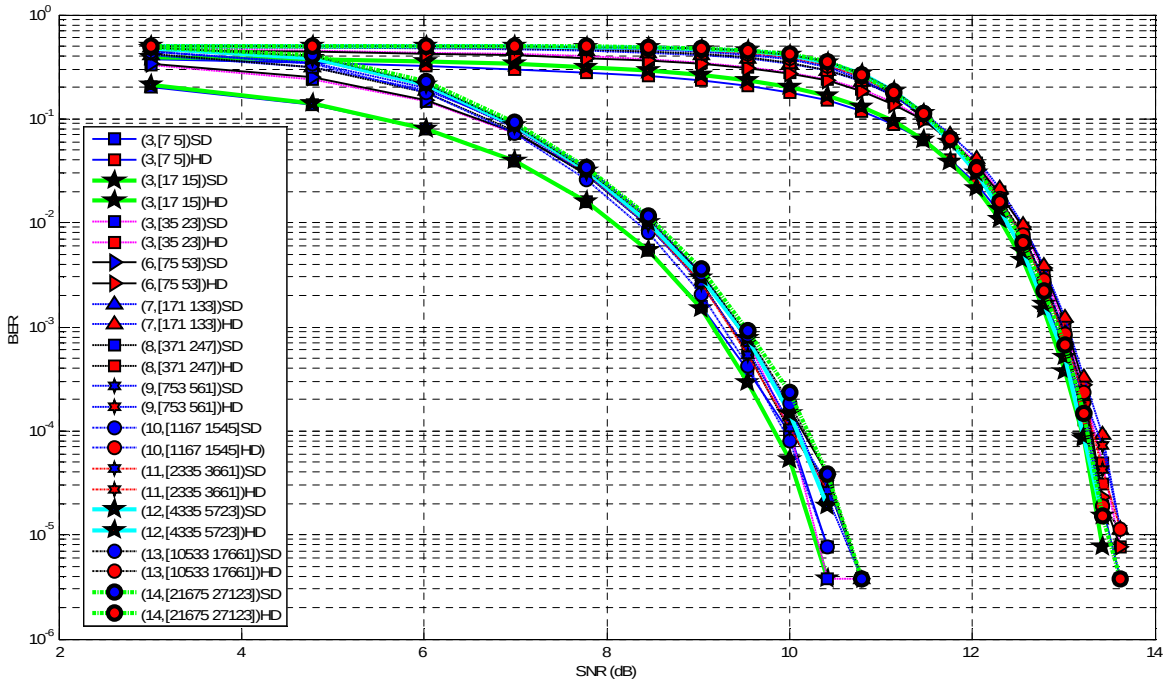
9. 16 Complexity Performance: Best Rate-1/2 Feedback-Controlled Hyperchaotic Encryption (D = 1.2), Chaotic Encryption (D = −1.2)

In this section we provide a more tangible comparison of complexity information and BER performance from several encoders developed for chaotic and hyperchaotic states generated by varying the control parameters in a feedback-controlled system analyzed in chapter 6. Several sets of closed-loop simulations are performed to demonstrate the benefit of information-based chaos system. In particular, it is demonstrated that two varieties of an information-based systems provide improved performance over all the encoder choices when hyperchaos states are utilized. The simulation parameters are set to A=36, B=3 and C=20. To generate a hyperchaotic (chaotic) data stream we choose D = 1.2 (D = −1.2). The corresponding positive Lyapunov exponents are

$\lambda_1^{(D=1.2)} = 1.402$, $\lambda_2^{(D=1.2)} = 0.529$ and $\lambda_1^{(D=-1.2)} = 0.997$. Each asymptotic system state is calculated by integration over 10^5 iterations after initial removal of 2×10^4 transient responses, with integration time-step 0.001 starting from initial conditions $(x_1, x_2, x_3, x_4) = (0.5, 1, -1, 1.2)$. It can be seen from Figure 9.52 (a) and (b) that the soft decision decoder has a better bit error performance. The decoder is able to recover the encrypted data and is able to reasonably estimate the bit error rate for different signal strengths under a noisy AWGN channel. This indicates that the encoder can update the information map in real time to compensate for changing data for both chaotic and hyperchaotic states. This is the evidence that occasional changes in the data stream can be handled by the decoder in a real time application. Samples of encoded data are shown in Figure 9.53 for $C_1^{(1/2)}(3,[5\ 7])$ and $C_{12}^{(1/2)}(14,[21675\ 27123])$ and decoded results are given in Figure 9.54 when hyperchaotic and chaotic states are used to encrypt and decrypt the data.

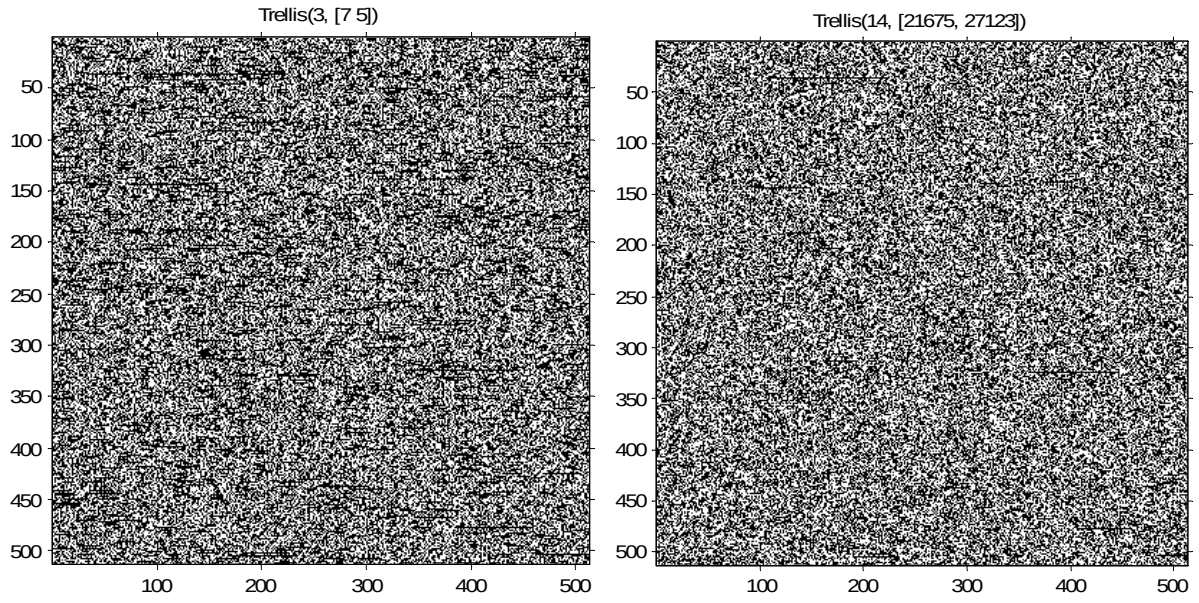


(a)

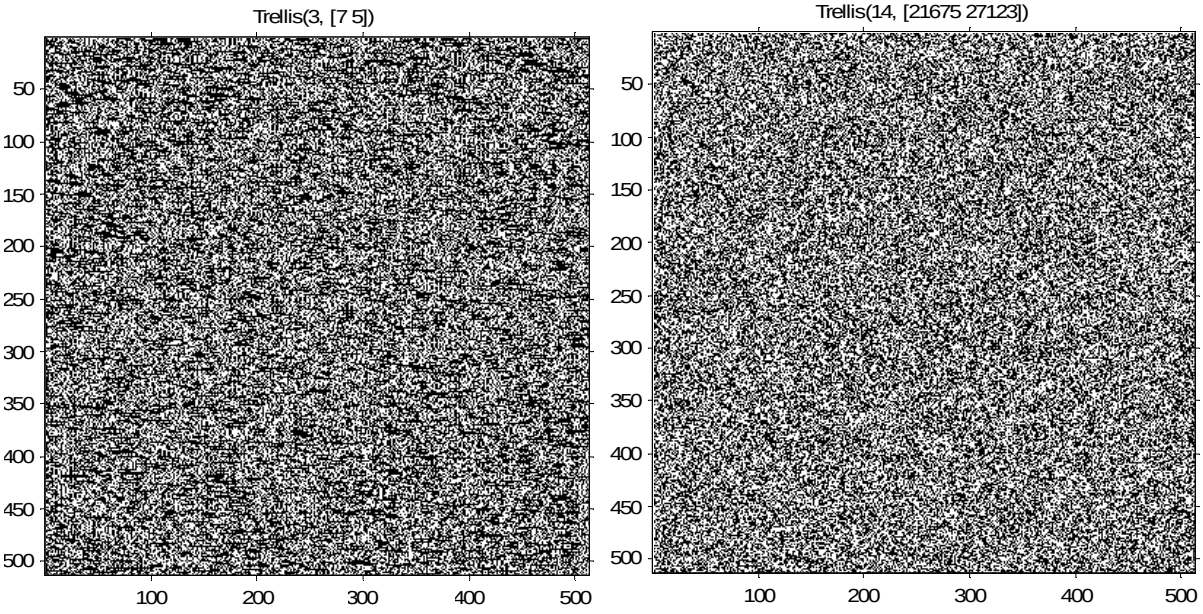


(b)

Figure 9.52 Soft decision and hard decision Viterbi decoder bit error rate performance on AWGN channel with 4-dim feedback controlled encryption (A=36, B=3, C=20) for best rate-1/2 convolutional codes: $C_1^{(1/2)}$ (3, [5 7]); $C_2^{(1/2)}$ (4, [15 17]); $C_3^{(1/2)}$ (5, [23 35]); $C_4^{(1/2)}$ (6, [53 75]); $C_5^{(1/2)}$ (7, [133 171]); $C_6^{(1/2)}$ (8, [247 371]); $C_7^{(1/2)}$ (9, [561 753]); $C_8^{(1/2)}$ (10, ,[1167 1545]); $C_9^{(1/2)}$ (11,[2335 3661]); $C_{10}^{(1/2)}$(12,[4335 5723]); $C_{11}^{(1/2)}$(13,[10533 17661]); $C_{12}^{(1/2)}$(14,[21675 27123]). (a) Hyperchaotic encryption (D = 1.2) (b) Chaotic encryption (D = −1.2).

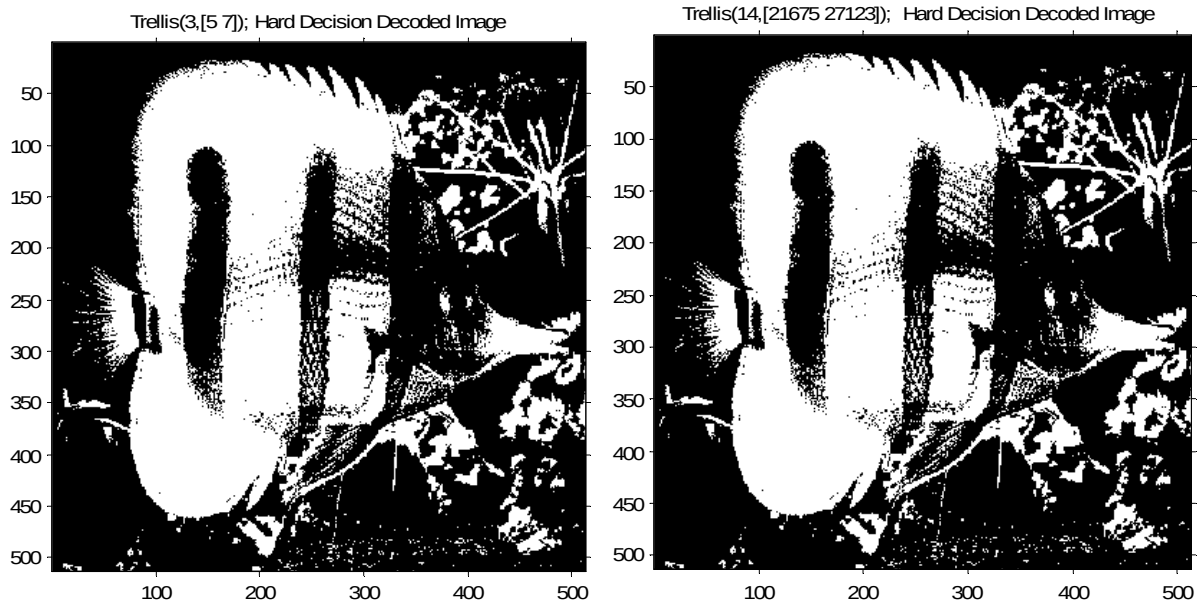


(a)

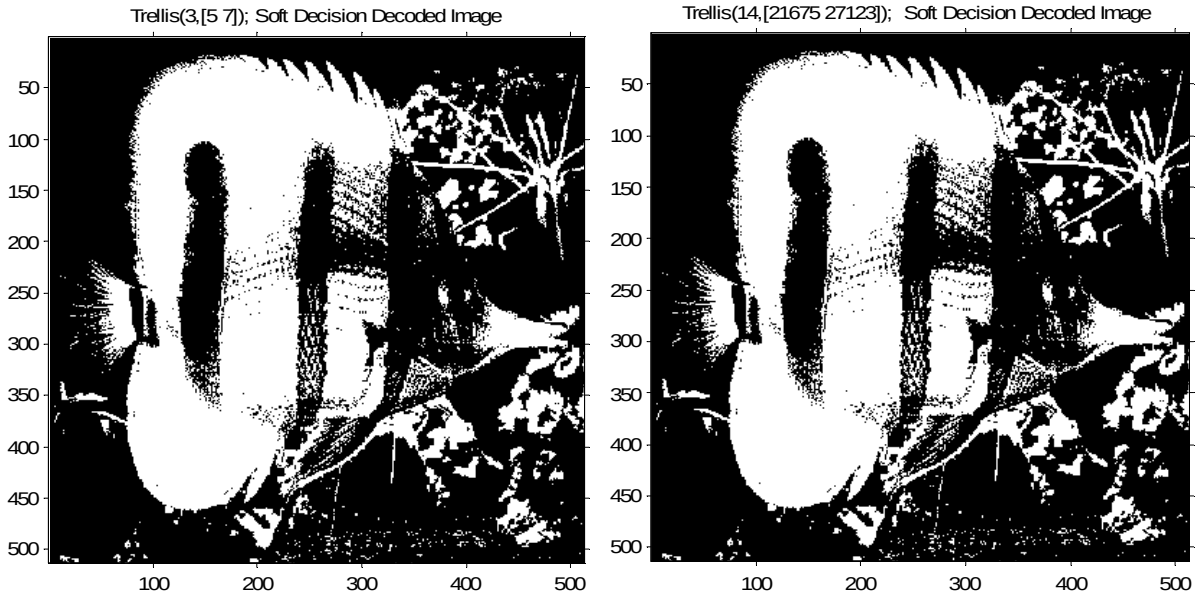


(b)

Figure 9.53 (a) Hyperchaotic encrypted (b) chaotic encrypted.



(a)



(b)

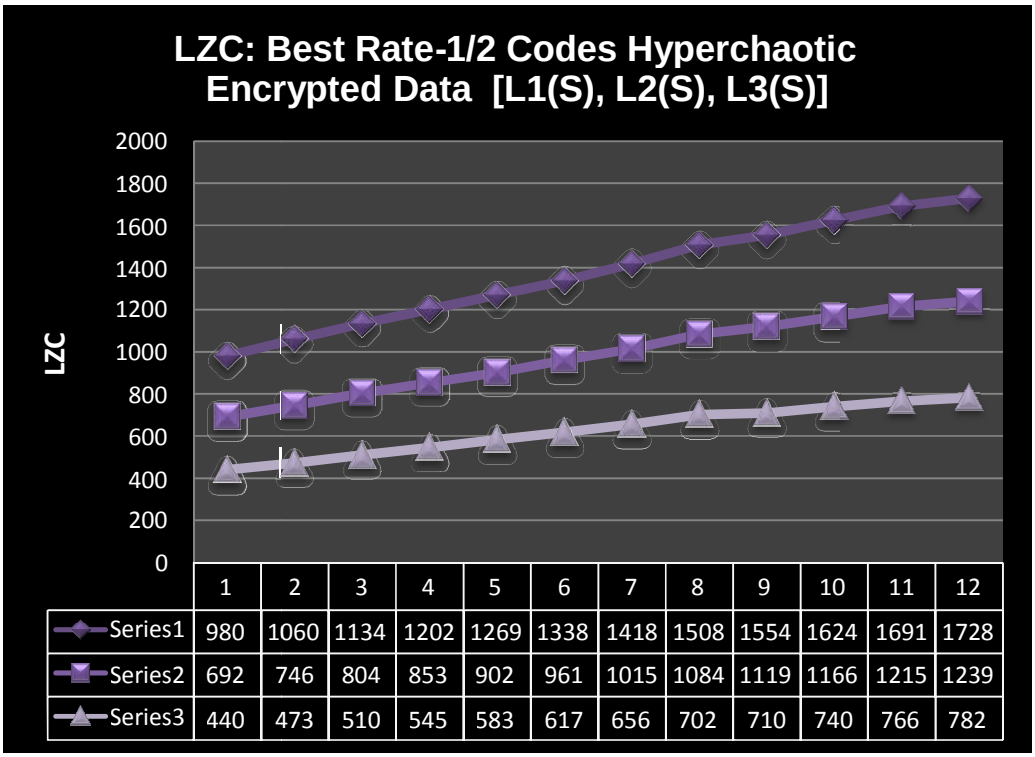
Figure 9.54 $C_1^{(1/2)}(3,[5\ 7])$, $C_{12}^{(1/2)}(14,[21675\ 27123])$ (a) Hyperchaotic decrypted (b) chaotic decrypted.

Table 9.14 Best rate-1/2 convolutional codes and their complexity measures and weight structure with feedback-controlled hyperchaotic and chaotic encrypted image.

Best Rate-1/2 codes	**$L_1(S)$ HyperChaos/ Chaos**	**$L_2(S)$ HyperChaos/ Chaos**	**$L_3(S)$ HyperChaos/ Chaos**	**Δ_{AVE} LZC per 10^4**	**d_{free}**
$C_1^{(1/2)}$(3, [5 7])	980/770	692/524	440/321	270/224	5
$C_2^{(1/2)}$(4, [15 17])	1060/795	746/538	473/327	293/234	6
$C_3^{(1/2)}$(5, [23 35])	1134/837	804/562	510/346	312/245	7
$C_4^{(1/2)}$(6, [53 75])	1202/880	853/588	545/366	328/257	8
$C_5^{(1/2)}$(7, [133 171])	1269/936	902/631	583/395	343/270	10
$C_6^{(1/2)}$(8, [247 371])	1338/979	961/658	617/419	360/280	10
$C_7^{(1/2)}$(9, [561 753])	1418/1037	1015/702	656/452	381/292	12
$C_8^{(1/2)}$(10, ,[1167 1545])	1508/1112	1084/758	702/494	403/309	12
$C_9^{(1/2)}$(11,[2335 3661])	1554/1153	1119/790	710/520	422/316	14
$C_{10}^{(1/2)}$(12,[4335 5723])	1624/1195	1166/826	740/540	442/327	15
$C_{11}^{(1/2)}$(13,[10533 17661])	1691/1251	1215/863	766/562	462/344	16
$C_{12}^{(1/2)}$(14,[21675 27123])	1728/1279	1239/889	782/583	473/348	16

As before, the information table is computed after the simulation. The information table is Table 9.14. This information table has two sets of information regions as opposed to the previous simulation. This is due to the information table computation using a chaos with hyperchaos state instead of just one chaos state. This table provides a comparison of the class $C^{(1/2)}$ encoders based on length, data stream types, LZ complexity, ΔLZC_{AVE} per 10^4 and codes free distance. 12 different encoders from the $C^{(1/2)}$ class across three trials with different sequence lengths are used in both hyperchaotic and chaotic states. Once the LZC is computed, the ΔLZC_{AVE} per 10^4 values are determined. For each computation scheme, two trials are run, and various LZC results are recorded. Each simulation scheme and the corresponding ΔLZC_{AVE} per 10^4 is estimated. The results are shown in Figures 9.55 and 9.56 for hyperchaos and chaos states respectively. Figure 9.57 reinforces that the hyperchaotic encoded state provide better

algorithmic complexity compared to the chaos state. This is also mirrored in $\Delta LZC_{AVE}(S)$ variations for 1/2-Rate maximum free distance class codes which are depicted in Figure 9.58. This property of the hyperchaotic state provides the sense that the high complex information codes have a smaller chance of unauthorized decryption.



	1	2	3	4	5	6	7	8	9	10	11	12
Series1	980	1060	1134	1202	1269	1338	1418	1508	1554	1624	1691	1728
Series2	692	746	804	853	902	961	1015	1084	1119	1166	1215	1239
Series3	440	473	510	545	583	617	656	702	710	740	766	782

Figure 9.55 LZC for the best rate-1/2 with data lengths $L_1(S)$, $L_2(S)$, $L_3(S)$ for a hyperchaotic encrypted data.

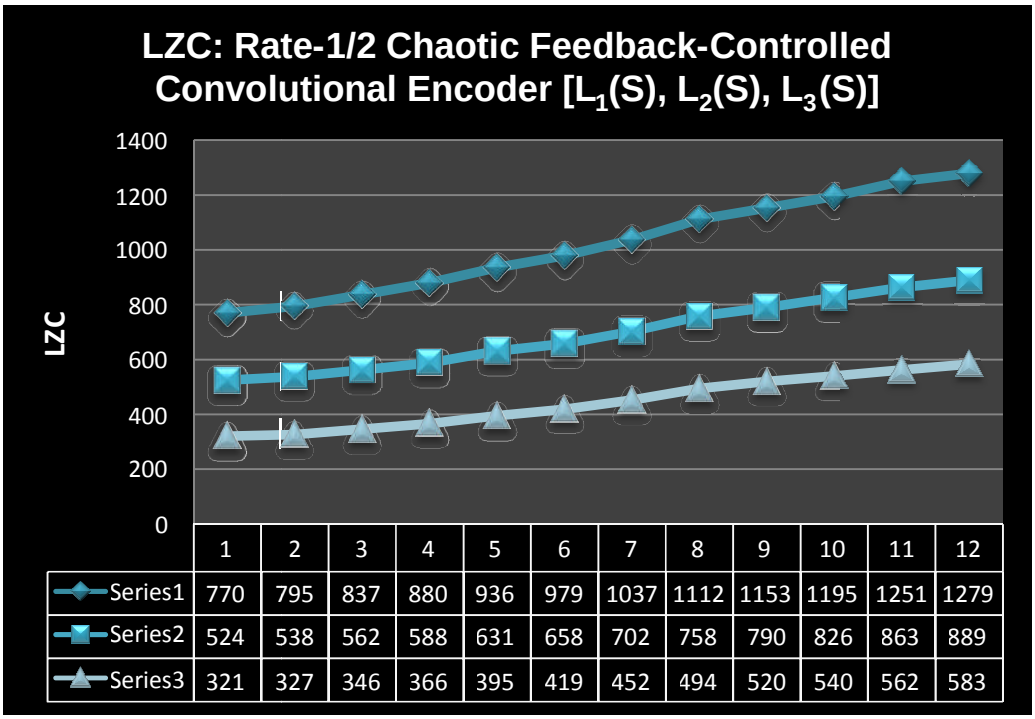


Figure 9.56 LZC for best rate-1/2 with data lengths $L_1(S)$, $L_2(S)$, $L_3(S)$ for a chaotic encrypted data.

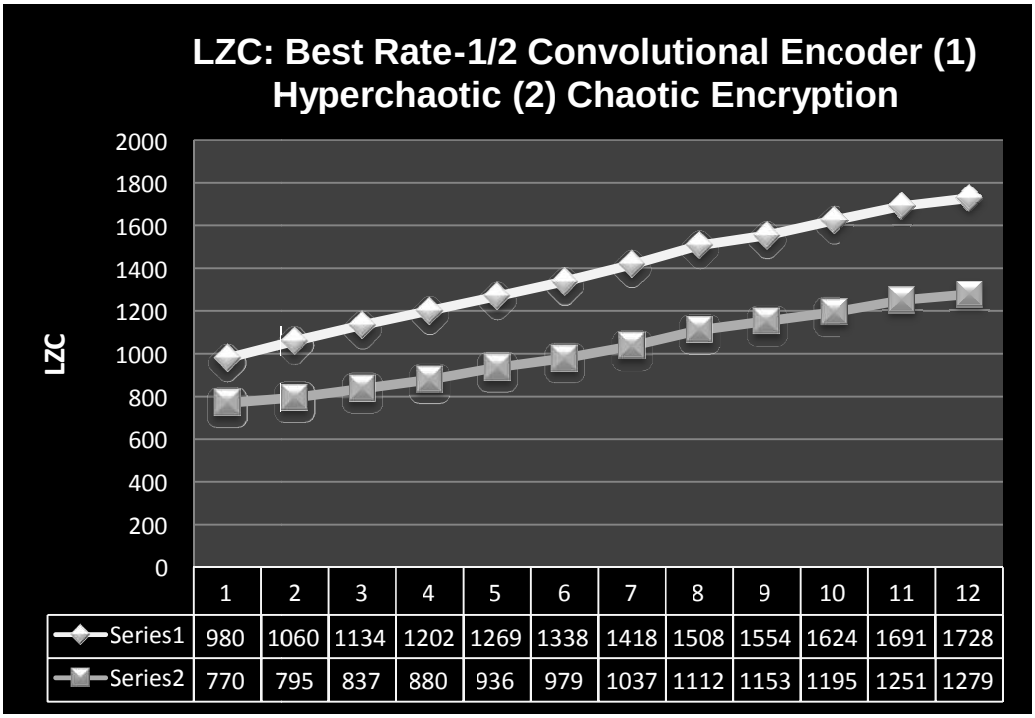


Figure 9.57 LZC comparisons of a hyperchaotic and chaotic encrypted data with the best rate-1/2 encoders.

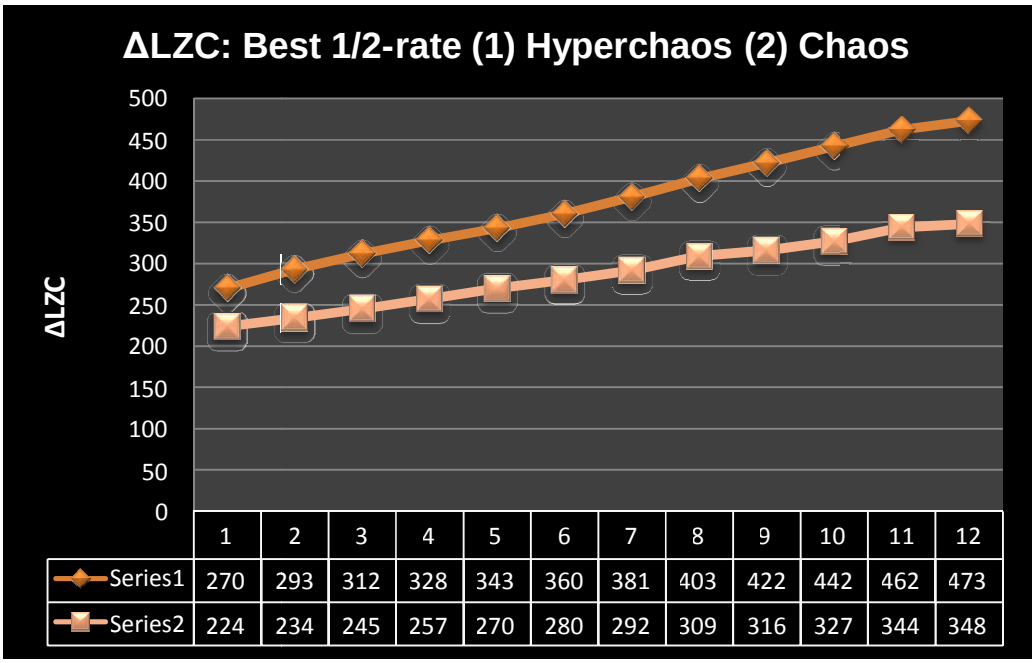
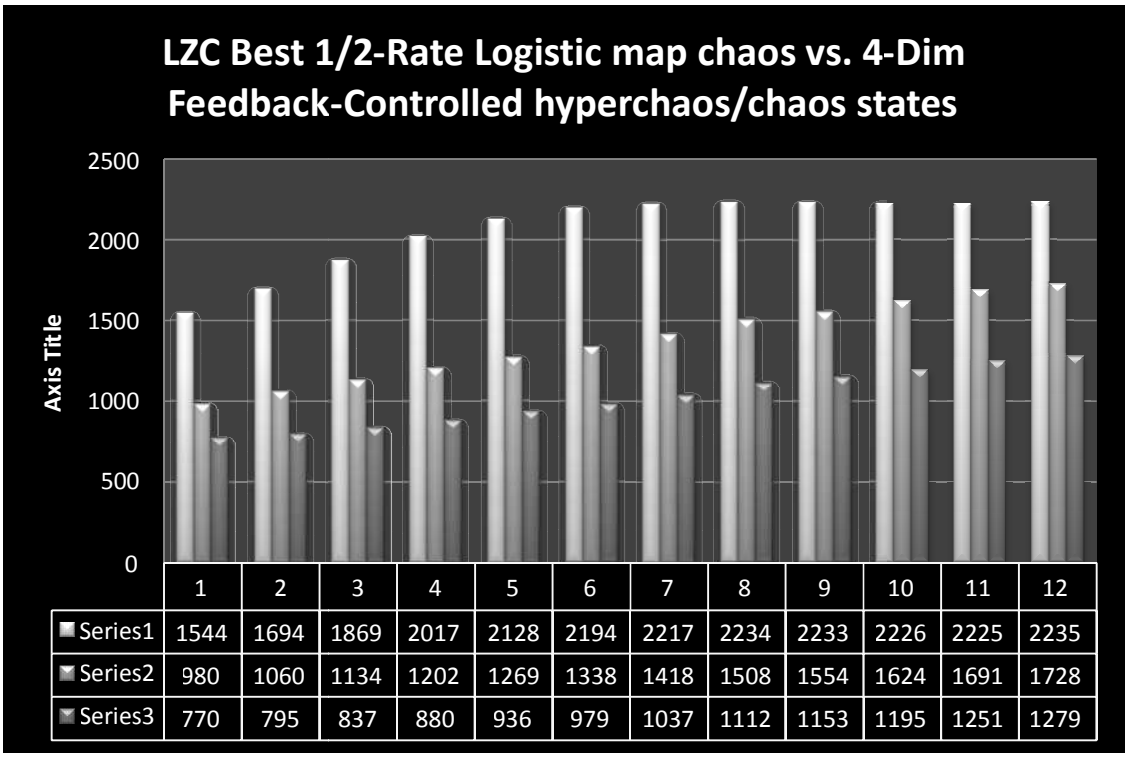


Figure 9.58 **Δ**LZC comparisons of a hyperchaotic and chaotic encrypted data with the best 1/2-rate encoders.

Finally, to observe the level of LZ complexity in three cases involving Logistic map chaotic states and 4-dim feedback-controlled hyperchaotic and chaotic responses the simulator is configured to emulate the $C^{(1/2)}$ class codes with the simulation parameters set to $A = 36$, $B = 3$, $C = 20$; (hyperchaos $D = 1.2$, chaos $D = -1.2$) for a 4-dim feedback-controlled system and $\mu = 0.4$ for the logistic map. The corresponding positive Lyapunov exponents are $\lambda_1^{(D = 1.2)} = 1.402$, $\lambda_2^{(D = 1.2)} = 0.529$, $\lambda_1^{(D = -1.2)} = 0.997$, $\lambda_1^{(\mu = 0.4)} = 0.693$. Observe that, unexpectedly, the logistic map scheme maintains a higher crypto-coded data stream complexity at all times than any other scheme as shown in Figure 9.59. In particular, the $C_{12}^{(1/2)}(14,[21675\ 27123])$ code has achieved the best complexity estimation performance. This scheme has 23% more LZC than the hyperchaotic scheme and 43% more LZC than the chaotic scheme in a 4-dim controlled-feedback system, even though the logistic map first Lyapunov exponent is smaller than that of the feedback controlled system. This is because of the redundant series of zeros (ones) in the

chaotic and hyperchaotic data streams generated by the feedback-controlled system. It is clear that by applying the advance sampling technique the optimum performance in complexity generation can be achieved. One major advantage of feedback-controlled systems is that they can be realized with op-amp electronic circuits which make them more attractive in hardware implementation of crypto-coded applications.



	1	2	3	4	5	6	7	8	9	10	11	12
Series1	1544	1694	1869	2017	2128	2194	2217	2234	2233	2226	2225	2235
Series2	980	1060	1134	1202	1269	1338	1418	1508	1554	1624	1691	1728
Series3	770	795	837	880	936	979	1037	1112	1153	1195	1251	1279

Figure 9.59 LZC comparisons of hyperchaotic and chaotic crypto-coded data. Logistic map and the 4-dim feedback-controlled system with best rate-1/2 convolutional encoders.

9.17 Conclusion

This chapter presents an overview of convolutional encoders with chaotic encrypted data in a secure digital communication system. Convolutional codes and Viterbi decoders are reviewed analytically. We define what constitutes the algorithmic complexity analysis for a

convolutional encoded chaotic-encrypted message. This work directly builds on the general idea of algorithmic complexity of data sequences generated from a cascade of a one-stage chaotic cryptosystem using a masking technique and a one-stage convolutionally encoding mechanism. We have designed and applied a simulating program that evaluates the efficiency and performance of this mechanism. One proposed application is for transmitting data on top of a non-guarantee protocol through the internet for provision efficient and secure real-time and multi-media applications. The cryptosystem is designed based on chaotic states. The output signal can be used to construct an unpredictable value for mixing with original data for encryption and decryption. Simulations for the various types of convolutional encoding are presented to determine the bit error rate for each type in conjunction with the use of the encoding strategy. This method can be used to create higher complexity by applying multi-stage chaotic encrypting and/or concatenating convolutional encoder units for a specific application. Special attention is paid to the algorithmic complexity of computing information for an entire class of rate-1/*n* encoders. We have measured the degree of diffusion in random-like patterns. Numerical evidence indicates the algorithmic complexity associated with particular 1/*n*-rate convolutional encoders increases as constraint length increases. This occurs in parallel with the increase of error correcting capacity of the decoder or free distance. This proves that algorithmic complexity is a suitable measure of the quality factor along with other measures of the weight structure of the code. Concerning a group of various rate-1/*n* convolutional encoders, we find that as the number of output streams, *n*, increases from 2 (or 1/*n* decreases from 1/2 to 1/7 rate codes) the free distance also increases while the encoded data sequence manifests low complexity in algorithm.

Finally, to observe the level of LZ complexity in three cases involving the logistic map chaotic states and 4-dim feedback-controlled hyperchaotic and chaotic responses the simulator is configured to emulate the $C^{(1/2)}$ class codes with the simulation parameters set to $A = 36$, $B = 3$, $C = 20$; (hyperchaos $D = 1.2$, chaos $D = -1.2$) for the 4-dim feedback-controlled system and $\mu = 0.4$ for the logistic map. The corresponding positive Lyapunov exponents are $\lambda_1^{(D = 1.2)} = 1.402$, $\lambda_2^{(D = 1.2)} = 0.529$, $\lambda_1^{(D = -1.2)} = 0.997$, $\lambda_1^{(\mu = 0.4)} = 0.693$. We observe that, unexpectedly, the logistic map scheme maintains higher crypto-coded data stream complexity at all times than any other scheme. In particular, the $C_{12}^{(1/2)}$(14,[21675 27123]) code has achieved the best complexity estimation performance. This scheme has 23% more LZC than the hyperchaotic scheme and 43% more LZC than the chaotic scheme in a 4-dim feedback-controlled system, even though the logistic map first Lyapunov exponent is smaller than that of the feedback-controlled system. This is because of the generation of a redundant long series of zeros (ones) in a slow dynamic feedback-controlled system. The system performance can be enhanced by applying the advance sampling technique.

REFERENCES

[1] Kannathal N, Sadasivan K Puthusserypady, Lim Choo Min. Complex dynamics of epileptic EEG. *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, September 2004.

[2] N. Hazarika, A. C. Tsoi, and A. A. Sergejew. Nonlinear considerations in EEG signal classification. *IEEE Transactions in Signal Processing*, Vol. 45, Apr. 1997.

[3] I. Yaylali, H. Kocak, and P. Jayakar. Detection of seizures from small samples using nonlinear dynamic system theory. *IEEE Transactions in Biomedical Engineering*, Vol. 43, pp. 743–751, July 1996.

[4] J. Jaeseung, C. Jeong-Ho, S. Y. Kim, H. Seol-Heui. Nonlinear dynamical analysis of the EEG in patients with Alzheimer's disease and vacular dementia. *Clinical Neurophysiology*, Vol. 18, No. 1, pp. 58-67, 2001.

[5] D. Lai, M. Palaniswami, *Computational Intelligence in Biomedical Engineering*, CRC press Dec. 2007.

[6] Pecora, L.M., and Carroll, T.L. Synchronization in chaotic systems. *Physical Review Letters*, Vol. 64, pp. 821-824, 1990.

[7] Hayes, S., Grebogi, C., and Ott, E. Communicating with chaos. *Physical Review Letters*, Vol. 70, pp. 3031-3034, 1993.

[8] F. Dachselt, W. Schwarz. Chaos and cryptography. *IEEE Transactions in Circuit and Systems*, Vol. 48, No. 12, pp. 1498-1509, Dec. 2001.

[9] G. Jakimoski and L. Kocarev, Chaos and cryptography: Block encryption ciphers based on chaotic maps. *IEEE Transactions in Circuits Sysems I, Fundam. Theory Appl.*, Vol. 48, No. 2, pp. 163–169, Feb. 2001.

[10] M. P. Kennedy, R. Rovatti, and G. Setti. *Chaotic Electronics in Telecommunications*. Boca Raton, FL: CRC Press, 2000.

[11] Blair Fraser, Pei Yu, and Turab Lookman. Steps towards improving the security of chaotic encryption. *Physical Review E,* Vol. 66, 017202, 2002.

[12] Raymond S. T. Lee and Henry W. S. Lam. A chaotic real-time cryptosystem using a switching algorithmic-based linear congruential generator (SLCG). *International Journal of Computer Science and Network Security*, Vol. 6, No.8 B, August 2006.

[13] D. Li, Y. Du. Artificial Intelligence with Uncertainty, *CRC press*, 2007.

[14] P. D. Wasserman. Advanced Methods in Neural Computing. Van Nostrand Reinhold, New York, 1993.

[15] E. N. Lorenz. Deterministic non-periodic flow. *J. Atmos. Sci*., Vol. 20, pp. 130-141, 1963.

[16] V. Petrov, V. Gaspar, J. Masere, and K. Showalter. Controlling chaos in the Belousov-Zhabotinsky reaction. *Nature*, Vol. 361, pp. 240-243, 1993.

[17] A. Lempel, J. Ziv. On the complexity of individual sequences. *IEEE Transactions in Information Theory*, Vol. IT-22, pp. 75-81, 1976.

[18] A. Lempel, J. Ziv. Compression of individual sequences via variable-rate coding. *IEEE Transactions in Information Theory*, Vol. IT-24, No. 5, pp. 530-536, 1978.

[19] F. Kasper, H.G. Shuster. Easily calculable measure for the complexity of spatiotemporal patterns. *Physical Review A*, Vol. 36, pp. 842-848, 1987.

[20] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano. Determining Lyapunov exponents from a time series. *Physica* D, Vol. 16, pp. 285–317, 1985.

[21] G. Rangarajan, S. Habib, R. D. Ryne. Lyapunov exponents without rescaling and reorthogonalization. *Physical Review A*, Vol. 36, pp. 832, 1998.

[22] W. Kinsner. Characterizing Chaos through Lyapunov Metrics. *IEEE Transactions on Systems, Man, and Cybernetics*, Part C, Vol. 36, No. 2, pp. 141-151, 2006.

[23] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge Univ. Press, 1997.

[24] P. Bryant, R. Brown, and H. D. I. Abarbanel. Lyapunov exponents from observed time series. *Physical Review Letters*, Vol. 65, pp. 1523–1526, 1990.

[25] J. Frøyland and K. H. Alfsen. Lyapunov exponent spectra for the Lorenz model. *Physical Review A*, Vol. 29, pp. 2928–2931, 1984.

[26] J. Frøyland, *Introduction to Chaos and Coherence*. Bristol, U.K. Institute of Physics, 1992.

[27] M. Sano and Y. Sawada. Measurement of the Lyapunov spectrum from a chaotic time series. *Physical Review Letters*, Vol. 55, pp. 1082–1085, 1985.

[28] J.-P. Eckmann, S. Oliffson-Kamphorst, D. Ruelle, and S. Ciliberto. Lyapunov exponents from time series. *Physical Review A*, Vol. 34, No. 6, pp. 4971–4979, 1986.

[29] A. G. Darbyshire and D. S. Broomhead. Robust estimation of tangent maps and Lyapunov spectra. *Physica D*, Vol. 89, pp. 287–305, 1996.

[30] J. C. Sprott, Chaos and Time-Series Analysis, Oxford University Press, 2003.

[31] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, 1997.

[32] P. Bryant. Computation of Lyapunov exponents from experimental data. *Proceedings of 1st Experimental Chaos Confenence.* S.Vohra, M. Spano, M. Shlesinger, L. Pecora, and W. Ditto, Eds. Singapore: World Scientific, pp. 11–23, 1992.

[33] U. Parlitz. Identification of true and spurious Lyapunov exponents from time series. *International Journal of Bifurcation and Chaos*, Vol. 2, pp. 155–165, 1992.

[34] D. Nychka, S. Ellner, A. R. Gallant, and D. McCaffrey. Finding chaos in noisy environments. *J. R. Statist. Soc. B*, Vol. 54, No. 2, pp. 399–426, 1992.

[35] Q. Yao and H. Tong. On prediction and chaos in stochastic systems. *Philos. Trans. R. Soc. A*, Vol. 348, No. 1688, pp. 357–369, 1994.

[36] B. D. Josephson. Possible new effects in superconductive tunneling. *Physics Letter* 1, p. 251, 1962.

[37] Niobium IC fabrication design, *www.hypres.com*

[38] E. Abraham, I. L. Atkin, and A. Wilson. Josephson voltage standard by controlling chaos. *IEEE Transactions on Applied Superconductivity*, Vol. 9, pp. 4166–4169, June 1999.

[39] W. C. Stewart. Current-voltage characteristics of Josephson junctions. *Applied Physics Letters*, Vol. 12, pp. 277–280, Apr. 1968.

[40] D. E. McCumber. Effect of ac impedance on dc voltage-current characteristics of superconductor weak-link junctions. *Journal of Applied Physics*, Vol. 39, pp. 3113–3118, June 1968.

[41] B.A. Huberman, J. P. Crutchfield, N. H. Packard. Noise phenomena in Josephson Junction. *Applied Physics Letters*, Vol. 37, 750, 1980.

[42] C. B. Whan and C. J. Lobb. Complex dynamical behavior in RCL shunted Josephson tunnel junctions. *Physical Review E*, Vol. 53, pp. 405–413, Jan. 1996.

[43] A. B. Cawthorne, C. B. Whan, and C. J. Lobb. Influence and evaluation of parasitic inductance in shunted Josephson junctions. *IEEE Transactions on Applied Superconductivity*, Vol. 7, pp. 2355–2358, June 1997.

[44] C. B. Whan and C. J. Lobb. Complex dynamical behavior in RCL shunted Josephson junctions. *IEEE Transactions on Applied Superconductivity*, Vol. 5, pp. 3094–3097, June 1995.

[45] J. A. Blackburn, H. J. T. Smith, and N. Gronbech-Jensen. Chaos and thermal noise in a Josephson junction coupled to a resonant tank. *Physical Review B*, Vol. 53, pp. 14546–14551, June 1996.

[46] M. Morisue, M. Yamadaya, H. Noguchi, and A. Kanasugi. Digital application of chaotic oscillation modes in Josephson circuit. *International Journal of Intelligent Systems*, Vol. 12, pp. 267–290, 1997.

[47] A. Kanasugi, M. Morisue, H. Noguchi, M. Yamadaya, and H. Furukawa. Oscillation modes in a Josephson circuit and its application to digital systems. *IEICE Transactions in Electronics*, Vol. E79-C, pp. 1206–1212, 1996.

[48] J.-G. Ma and I. Wolff. The chaotic oscillations of a Josephson junction with external magnetic field. *IEEE Transaction in Magnetism*, Vol. 32, pp. 1164–1167, May 1996.

[49] A. Takada, S. Kaechi, T. Furukawa, and T. Kuwahara. Range of chaotic state in the Josephson junction driven by a pulse train signal. *IEEE Transactions on Applied Superconductivity*, Vol. 9, pp. 3962–3965, June 1999.

[50] A. Kenfack and T. C. Kofane. Chaos in rf-driven long Josephson junctions in the presence of an external field. *Physical Review B*, Vol. 52, pp. 10359–10363, 1995.

[51] K. N. Yugay, N. V. Blinov, and I. V. Shirokov. Effect of memory and dynamical chaos in long Josephson junctions. *Physical Review B*, vol. 51, pp. 12737–12741, 1995.

[52] G. Filatrella and G. Rotoli. Long Josephson junctions driven by biharmonic signals. In *Physical Review B*, Vol. 50, pp. 12802–12810, 1994.

[53] H. How, T.-M. Fang, C. Vittoria, and A. Widom. Nonlinear mixer gain calculations for Josephson junctions. *IEEE Transaction in Microwave Theory Technology*, Vol. 43, pp. 216–218, 1995.

[54] J. L. Kaplan and J. A. Yorke. Chaotic behavior of multidimensional difference equations. *Functional Differential Equations and Approximation of Fixed Points*, H.-O. Peitgen and H. O. Walther, Eds. New York: Springer-Verlag, 1979, pp. 204–227.

[55] D. Arasteh, M. R. Kolahchi. LZ Complexity of the chaotic dynamical systems. *Iranian Journal of Physics Research*, Vol. 1, No.4, pp. 208–221, 1998.

[56] Y. B. Pesin. Characteristic Lyapunov exponents and smooth ergodic theory. *Russian Mathematical Surveys*, Vol. 32, pp. 55–114, 1977.

[57] S. Savari. Redundancy of the Lempel-Ziv incremental parsing rule. *IEEE Transactions in Information Theory*, Vol. 43, No. 1, pp. 9–21, 1997.

[58] P. Grassberger. Estimating the information content of symbol sequences and efficient codes. *IEEE Transaction in Information Theory*, Vol. 35, No. 3, pp. 669–675, 1989.

[59] A. Lapidoth and J. Ziv. On the universality of the LZ-based decoding algorithm. *IEEE Transactions in Information Theory*, Vol. 44, No. 5, pp. 1746–1755, 1998.

[60] G. Louchard and W. Szpankowski. On the average redundancy rate of the Lempel-Ziv code. *IEEE Transactions in Information Theory*, Vol. 43, No. 1, pp. 2–8, 1997.

[61] T. Luczak and W. Szpankowski. A suboptimal lossy data compression based on approximate pattern matching. *IEEE Transactions in Information Theory*, Vol. 43, No. 5, pp. 1439–1451, Sep. 1997.

[62] D. Ornstein and B. Weiss. Entropy and data compression schemes. *IEEE Transactions in Information Theory*, Vol. 39, No. 1, pp. 78–83, Jan. 1993.

[63] A. Gavish and A. Lempel. Match-length functions for data compression. *IEEE Transactions in Information Theory*, Vol. 42, No. 5, pp. 1375–1380, Sep.1996.

[64] E.-H. Yang and J. Kieffer. Simple universal lossy data compression schemes derived from the Lempel-Ziv algorithm. *IEEE Transactions in Information Theory*, Vol. 42, No. 1, pp. 239–245, Jan. 1996.

[65] A. Lempel and M. Cohn. Design of universal test sequences for VLSI. *IEEE Transactions in Information Theory*, Vol. IT-31, No. 1, pp. 10–17, Jan. 1985.

[66] K. Visweswariah, S. Kulkarni, and S. Verdu. Source codes as random number generators. *IEEE Transactions in Information Theory*, Vol. 44, No. 2, pp. 462–471, 1998.

[67] T. Etzion and A. Lempel. Algorithms for the generation of full-length shift-register sequences. *IEEE Transactions in Information Theory*, Vol. IT-30, No. 3, pp. 480–484, May 1984.

[68] M. Aboy, R. Hornero, D. Abásolo, D. Álvarez. Interpretation of the Lempel-Ziv Complexity Measure in the Context of Biomedical Signal Analysis. *IEEE Transactions in Biomedical Engineering*, Vol. 53, No. 11, pp. 2282–2288, Nov. 2006

[69] H. H. Otu and K. Sayood. A new sequence distance measure for phylogenetic tree construction. *Bioinformatics*, Vol. 19, No. 16, pp. 2122–2130, Nov. 2003.

[70] J. Xu, Z. Liu, R. Liu, and Q. F. Yang. Information transformation in human cerebral cortex. *Physica D*, Vol. 106, pp. 363–374, 1997.

[71] D. Abásolo, R. Hornero, C. Gómez, M. García, and M. López. Analysis of EEG background activity in Alzheimer's disease patients with Lempel-Ziv complexity and central tendency measure. *Medical Engineering and Physics*, Vol. 28, No. 4, pp. 315–322, May 2006.

[72] N. Radhakrishnan and B. Gangadhar. Estimating regularity in epileptic seizure time-series data. *IEEE Engineering in Medicine and Biology Magazine*, Vol. 17, No. 3, pp. 89–94, May-Jun. 1998.

[73] X. S. Zhang and R. J. Roy. Predicting movement during anesthesia by complexity analysis of electroencephalograms. *Medical Biology and Engineering Computing*, Vol. 37, No. 3, pp. 327–334, May 1999.

[74] X. S. Zhang, R. J. Roy, and E. W. Jensen. EEG complexity as a measure of depth of anesthesia for patients. *IEEE Transactions in Biomedical Engineering*, Vol. 48, No. 12, pp. 1424–1433, Dec. 2001.

[75] X. S. Zhang and R. J. Roy. Derived fuzzy knowledge model for estimating the depth of anesthesia. *IEEE Transactions in Biomedical Engineering*, Vol. 48, No. 3, pp. 312–323, Mar. 2001.

[76] J.W. Zhang and C. X. Zheng. EEG complexity measurement of focal ischemic cerebral injury. *Proceedings of the 20th Annual International Conference, IEEE Engineering in Medicine and Biology Society*, Vol. 4, pp. 2027–2029, Hong Kong, China, 1998.

[77] R. Nagarajan. Quantifying physiological data with Lempel-Ziv complexity-certain issues. *IEEE Transactions in Biomedical Engineering*, Vol. 49, No. 11, pp. 1371–1373, Nov. 2002.

[78] Garfinkel, A., Spano, M. L., Ditto, W. L., and Weiss, J. A. Controlling cardiac chaos. *Science*, Vol. 257, pp. 1230-1235, 1992.

[79] X. S. Zhang, Y. S. Zhu, and X. J. Zhang. New approach to studies on ECG dynamics: extraction and analyses of QRS complex irregularity time series. *Medical Biology and Engineering Computing*, Vol. 35, No. 5, pp. 467–473, Sep. 1997.

[80] X. S. Zhang, Y. S. Zhu, N. V. Thakor, and Z. Z. Wang. Detecting ventricular tachycardia and fibrillation by complexity measure. *IEEE Transactions in Biomedical Engineering*, Vol. 46, No. 5, pp. 548–555, May 1999.

[81] H. X. Zhang, Y. S. Zhu, and Z. M. Wang. Complexity measure and complexity rate information based detection of ventricular tachycardia and fibrillation. *Medical Biology and Engineering Computing*, Vol. 38, No. 5, pp. 553–557, Sep. 2000.

[82] Y. L. Orlov and V. N. Potapov. Complexity: An internet resource for analysis of DNA sequence complexity. *Nucleic Acids Research*, Vol. 32, pp. W628–W633, July 2004.

[83] V. D. Gusev, L. A. Nemytikova, and N. A. Chuzhanova. On the complexity measures of genetic sequences. *Bioinformatics*, Vol. 15, No. 12, pp. 994–999, Dec. 1999.

[84] L. Stern, L. Allison, R. L. Coppel, and T. I. Dix. Discovering patterns in plasmodium falciparum genomic DNA. *Molcular Biochemistray and Parasitology*, Vol. 118, No. 2, pp. 175–186, Dec. 2001.

[85] J. Szczepaski, J. M. Amigó, E. Wajnryb, and M. V. Sanchez-Vives. Application of Lempel-Ziv complexity to the analysis of neural discharges. *Network*, Vol. 14, No. 2, pp. 335–350, May 2003.

[86] Ruqiang Yan and Robert X. Gao. Complexity as a Measure for Machine Health Evaluation. *IEEE Transactions on Instrumentation and Measurement*, Vol. 53, No. 4, Aug. 2004.

[87] J. Sofia Hadjileontiadou et. al. On Efficiently Tracking Turn-Taking Patterns in a CSCL environment using Lempel-Ziv Complexity Analysis. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 2004.

[88] J. Soto, Statistical testing of random number generators. *Proceedings of the 22nd National Information Systems Security Conference*, Crystal City, Virginia, 1999.

[89] A. Uchida et al. Chaotic Oscillations in Josephson Tetrode. *IEEE Transactions on Applied Superconductivity*, Vol. 14, No. 4, pp. 2064–2070, Dec. 2004.

[90] V. G. Ivancevic, T. T. Ivacrvic. High-Dimentional Chaotic and Attractor Systems. Springer 2007.

[91] G. D. Van Wiggeren and R. Roy. Communication with chaotic lasers. In *Science*, Vol. 279, pp. 1198–1200, Feb. 1998.

[92] J. P. Goedgebuer, L. Larger, and H. Porte. Optical cryptosystem based on synchronization of hyperchaos generated by a delayed feedback tunable laser diode. *Physical Review Letters*, Vol. 80, pp. 2249–2252, Mar. 1998.

[93] H. How, T.-M. Fang, C. Vittoria, and A. Widom. Nonlinear mixer gain calculations for Josephson junctions. *IEEE Transactions in Microwave Theory and Technology*, Vol. 43, pp. 216–218, Jan. 1995.

[94] H. Xu, M. Fan, S. Yang, and P. Wu. Chaos in superconducting tunnel junctions. *Journal of Applied Physics*, Vol. 78, pp. 6664–6669, Dec. 1995.

[95] O.E. Rossler. An equation for hyperchaos. *Physics Letters A*, Vol. 71, p. 155, 1979.

[96] C.Z. Ning, H. Haken. Detuned lasers and the complex Lorenz equations: Subcritical and supercritical Hopf bifurcations. *Physical Review A*, Vol. 41, pp. 3826-3837, 1990.

[97] T. Kapitaniak, L.O. Chua. Hyperchaotic attractors of unidirectionally coupled Chua's circuits. *International Journal of Bifurcation and Chaos*, Vol. 4, pp. 477-483, 1994.

[98] Simin Yu, Jinhu Lü, Wallace K. S. Tang, Guanrong Chen. A general multiscroll Lorenz system family and its realization via digital signal Processor. *Chaos*, Vol. 16, 033126, 2006.

[99] A. S. Elwakil, S. Ozoguz. On the generation of higher order chaotic oscillators via passive coupling of two identical or nonidentical sinusoidal oscillators. *IEEE Transactions in Circuits and Systems*, Vol. 53, No. 7, July 2006.

[100] L. Kocarev, K.S. Halle, K. Eckert and L.O. Chua. Experimental demonstration of secure communication via chaotic synchronization. *International Journal of Bifurcation and Chaos*, Vol. 2, pp. 709-713, 1992.

[101] M. Itoh. Spread spectrum communication via chaos. *International Journal of Bifurcation and Chaos*, Vol. 9, pp. 155-213, 1996.

[102] K.Murali, H. Yu, V. Varadan and H. Leung. Secure communication using a chaos based signal encryption scheme. *IEEE Transactions in Consumer Electronics*, Vol. 47, pp. 709-714, 2001.

[103] K. Murali, H. Leung, K. Shakthi Preethi and I. Raja Mohamed. Spread spectrum image encoding and decoding using ergodic chaos. In *IEEE Transactions in Consumer Electronics*, Vol. 49, No.1, pp. 59-63, 2003.

[104] K.M. Cuomo and A.V. Oppenheim. Circuit implementation of synchronized chaos with applications to secure communication. *Physics Letters A*, Vol. 71, pp. 65-68, 1998.

[105] M.P. Kennedy and G. Kolumban. Digital communications using chaos. *Signal Processing*, Vol. 80, pp. 1307-1320, 2000.

[106] H. Dedieu, M.P. Kennedy and M. Hasler. Chaos shift-keying: Modulation and demodulation of a chaotic character using self synchronizing Chua's circuits. *IEEE Transactions in Circuits & System-II*, Vol. 40, pp. 634-642, 1993.

[107] K. Murali. Digital signal transmission with cascaded heterogeneous chaotic systems. *Physical Letters A*, Vol. 63, pp. 016217-016223, 2001.

[108] A.S. Elwakil and M.P. Kennedy. A low-voltage, low-power, chaotic oscillator, derived from a relaxation oscillator. *Microelectronics Journal*, Vol. 31, pp. 459-468, 2000.

[109] S.E. de S. Pinto, S.R. Lopes and R.L. Viana. Collective behavior in a chain of Van der Pol oscillators with power-law coupling. *Physica A*, Vol. 303, pp. 339-356, 2002.

[110] J. Kurths, S. Boccaletti, C. Grebogi and Y.-C. Lai. Introduction: Control and synchronization in chaotic dynamical systems. *Chaos*, Vol. 13, pp. 126-127, 2003.

[111] M.G. Rosenblum, A.S. Pikovsky and J. Kurths. Phase Synchronization of Chaotic Oscillators, *Physical Review Letters*, Vol. 76, pp. 1804-1807, 1996.

[112] M. G. Rosenblum, A. Pikovsky and J. Kuths. *Physical Review Letters*, Vol. 78, pp. 4193-4196, 1997.

[113] C. Shannon. Communication Theory of Security Systems. *Bell Systems Tech J*., Vol. 28, Oct. 1949.

[114] S. Panwar, S. Mao, J. Ryoo, Y. Li. TCP/IP Essentials: A Lab-Based Approach. Cambridge University Press, 2004.

[115] D. Boneh and J. Shaw. Collusion-Secure Fingerprinting for Digital Data. *Advances in*

Cryptology - CRYPTO 95, Lecture Notes in Computer Science, Springer-Verlag, Berlin 1995.

[116] M. Fernandez and M. Soriano. Identification of Traitors in Algebraic-Geometric Traceability Codes. In *IEEE Transaction on Signal Processing, Supplement on Secure Media*, Vol. 52, No. 10, pp. 3073–3077, 2004.

[117] Yan Zhu, Wei Zou. Collusion Secure Convolutional Fingerprinting Information Codes. *ASIACCS'06*, Taiwan, pp. 266-274, 2006 ACM.

[118] R. Prasad and T. Ojanpera. An Overview of CDMA Evolution toward Wideband CDMA. *IEEE Communications Surveys*, Vol. 1, No. 1, Fourth Quarter, pp. 2-29, 1998.

[119] F. Chan and D. Haccoun, Adaptive Viterbi Decoding of Convolutional Codes over Memoryless Channels. *IEEE Transactions on Communications*, Vol. 45, No. 11, pp. 1389-1400, 1997.

[120] A. R. Calderbank. The art of signaling: Fifty years of coding theory. *IEEE Transactions Information Theory*, Vol. 44, pp. 2561-2595, Oct. 1998.

[121] Alexander Vardy. Algorithmic Complexity in Coding Theory and the Minimum Distance Problem. *STOC'97* El Paso, Texas, pp. 92-109, 1997 ACM.

[122] J. Proakis. *Digital Communication*, Fifth edition. McGraw-Hill, 2008.

[123] A. Chen, J. Lu, J. Lu, S. Yu. Generating hyperchaotic Lu attractor via state feedback control. *Physica* A, Vol. 364, pp. 103-110, 2006.

VITA

Davoud Arasteh was born in Tehran, Iran. He received his Bachelor and Master's degrees in Applied Physics from the National University of Iran, and Amir Kabir University of Technology (Tehran Polytechnic University), in 1991 and 1994. He enlisted in the army for his mandatory military service. He served for two years as a first lieutenant. After his discharge he left Iran to pursue graduate studies in Telecommunication Engineering at the University of Louisiana, Lafayette. He received his Master's degree from ULL in 2002. He pursued his doctoral studies in Louisiana State University System at the University of New Orleans. He has been a faculty member at Southern University since 1999, serving the Electrical Engineering and Electronic Engineering Technology Departments in which he has taught by developing curricula, participating in faculty development activities and implementing new teaching technologies in computer networking and secure computing. His research interests include Dependable and Secure Computing, Computational Intelligence, Engineering Applications of Nonlinear Systems, Quantum Computing and Bioinformatics.