# TARGETING VIA REGION SCHEDULING FOR THE CONTROL OF CHAOTIC SYSTEMS

by

Özgür AKSOY

B.S. in MATH., Istanbul Technical University, 1995

B.S. in CMPE., Istanbul Technical University, 1995



Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Electronics Engineering

Boğaziçi University

2000

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Assoc. Prof. Yağmur Denizhan for her endless support, constructive supervision and patience. In all phases of this study, her friendly and wise attitudes have been the source of my motivation. Special thanks to her.

Many thanks to Prof. Okyay Kaynak and Prof. Fikret Gürgen for being in my thesis jury.

I am also indebted to my friends M. Önder Efe, Serdar İplikçi, Mustafa Güler and Ali Yapar for their comments, suggestions and enthusiastic responses.

Above all, I would like to express my appreciations to the one, whose incredible support has always been in my mind.

# ABSTRACT

The local control method for chaotic dynamics as proposed by Ott, Grebogi and Yorke (the OGY method) has drawn the attention of many non-linear system researchers within the last decade. This method exploits the properties of chaotic dynamics for an acceptable control performance. The OGY method can stabilise a target, i.e. a chosen unstable equilibrium point or an unstable periodic orbit, without a priori knowledge about the system dynamics. In the literature there exist many extensions and modifications of this approach. The major drawback of these methods is the long waiting time, which is required the system converges to a close neighbourhood of the target since the control is based on a linearization around the target. In addition to the OGY method and its extensions, neural networks are also being utilised for the control of chaotic dynamics.

The aim of this work is to achieve the stabilization of the target without priori knowledge about the system dynamics, while reducing the average time to reach the close neighbourhood. The local modelling of the dynamics is achieved using a neural networks employing Radial Basis Functions. A sufficient reduction in the reaching time and satisfactory stabilization of the desired target has been achived by this method.

# ÖZET

Son on yılda, kaotik dinamikler için Ott, Grebogi ve Yorke tarafından önerilen lokal denetim yöntemi (OGY yöntemi) lineer olmayan sistemler konusunda çalışan pek çok araştırmacının ilgisini çekmiştir. Bu yöntem, kabul edilebilir bir denetim performası sağlamak için kaotik dinamiklerin bazı özelliklerini kullanır. OGY yöntemi bir hedefi, yani seçilmiş bir kararsız denge noktasını veya kararsız bir periyodik yörüngeyi sistemin denklemlerini gerektirmeden kararlı hale getirebilir. Literatürde bu yaklaşımın pek çok uyarlamaları mevcuttur. Denetimin lokal doğrusallaştırmaya dayanmasından dolayı bu yöntemlerin, sistemin hedef civarına gelene kadar beklemek zorunda olması onların en önemli dezavantajıdır. OGY yöntemi ve onun genişletilmiş versiyonlarına ek olarak yapay sinir ağları da kaos denetiminde halen kullanılmaktadır.

Bu çalışmanın amacı öncelikle, sistemin denklemlerini gerektirmeden hedefi kararlı hale getirirken, sistemin hedefe ortalama erişim zamanını azaltmaktır. Sistem dinamiklerinin lokal modellenmesi, radyal bazlı fonksiyonları kullanan bir yapay sinir ağı ile gerçeklenmiştir. Bu yöntemle, ortalama erişim zamanında yeterli bir azalma ve istenilen hedefte doyurucu bir kararlılık elde edilmiştir.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS

| | |
|---|---|
| $\underline{e}$ | The eigenvector of the linearized map |
| $\underline{f}$ | The contravariant basis vector of the linearized map |
| $K$ | Number of the regions on Poincaré surface |
| $\underline{p}$ | Parameter vector of a chaotic system |
| $\underline{p}_{nom}$ | Vector of the nominal values of the parameters |
| R | Real numbers space |
| $R_i$ | The $i^{th}$ region of the Poincaré surface |
| $\underline{x}$ | State vector of a continuous-time chaotic system |
| $\underline{z}$ | State vector of a discrete-time chaotic system |
| $\underline{x}^*$ | Equilibrium point vector of a continuous-time chaotic system |
| $\underline{z}^*$ | Equilibrium point vector of a discrete-time chaotic system |
| $\underline{\delta p}_{max}$ | Maximum allowable parameter vector |
| $\Pi$ | Allowable range of the control parameters |
| $\lambda$ | The eigenvalue of the linearized map |
| $\delta$ | Radius of the control region |

# 1. INTRODUCTION

In the last two decades scientists have begun to understand that very simple mathematical equations can produce quite complicated behaviour. *Chaos* is apparently unpredictable behaviour arising in a deterministic system because of great sensitivity to initial conditions. Deterministic systems are *predictable*: Given the initial conditions and the equations describing a system, its future behaviour can be predicted for all time. The chaotic behaviour in a deterministic system, also called the "*strange behaviour*", is unpredictable because of the great sensitivity to initial conditions. Chaos arises in a dynamical system if two arbitrarily close starting points diverge exponentially, so that their long-term behaviour is unpredictable. Such a system is also highly sensitive to any disturbance.

Chaotic behaviour is observed in various fields like weather, biology, economics, astronomy, medicine etc. Weather is one of the most investigated examples among systems, which exhibit chaotic behaviour. Forecasts are never totally accurate, and long-term forecasts, even for one week, can be totally wrong. This can be due to the unmodelled dynamics, however, even if the system dynamics are perfectly known, due to the exponentially diverging behaviour of nearby trajectories and the impossibility of measuring the initial conditions with infinite accuracy.

In recent years, the control of chaotic systems has become a focus in the study of non-linear dynamics. Generally speaking, to control a system means to make it exhibit a desired behaviour. Chaos control methods can in general be divided into two main categories: Firstly, feedback methods, in which the desired phase space trajectory is determined and some state or output feedback is employed to make the system track the desired trajectory, and secondly, control methods based on modification of control parameters, where some properties or knowledge of the system is used to modify the chaotic behaviour.

## 1.1. Basic Concepts of Non-Linear Dynamics and Chaos

A *dynamical system* is a system, whose evolution is deterministic in the sense that its future motion is determined by its current state. The knowledge of *state* contains the information upon which the dynamical system acts.

Two of the most common dynamic system representations involve differential and difference equations,

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t)] \tag{1.1.a}$$

$$\underline{x}_{n+1} = \underline{f}(\underline{x}_n) \tag{1.1.b}$$

In (1.1), $x$ represents the state vector of the dynamical system. Any differential or difference equation can be brought into the form of (1.1) by introducing the higher order derivatives as new state variables. A continuous-time system can be described as in equation (1.1.a), which is referred to as a *flow*. Similarly, a discrete-time system can be described as in equation (1.1.b), which is referred to as a *map*.

The dimension of the state vector is the *dimension* of the system.

The set of all possible states of a system is called the *phase space (or state space)* of the system. The evolution of a system starting from an initial state in the state space is referred to as a *phase trajectory*.

*Equilibrium points* of flows or maps are states on which the system stays forever once those are reached. *Periodic orbits* of flows are closed system trajectories, i.e. solutions of (1.1) satisfying (1.2).

$$\underline{x}(t+T) = \underline{x}(t) \tag{1.2}$$

where $T$ represents the period of the orbit. States of maps satisfying (1.3) are referred to as *period-p points*. It should be noted that the equilibrium points of maps can be considered as period-1 points. p successive period-p points constitute a *period-p orbit* for a map.

$$\underline{x}_{n+p} = \underline{x}_n \tag{1.3}$$

where $p$ represents the period of the orbit.

Isolated equilibrium points and periodic orbits of a system are referred to as *limit sets*.

*Stability* of limit sets can be determined from the eigenvalues of the Jacobian evaluated at the limit set under consideration (1.4).

$$\left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{x=x^*} \tag{1.4}$$

An equilibrium point (or periodic orbit) is defined to be *stable* if all sufficiently small disturbances away from it damp out in time. Conversely, if all sufficiently small disturbances grow in time, then an equilibrium point (or periodic orbit) is said to be *unstable*. If the effect of the disturbance depends on the direction then equilibrium point (or periodic orbit) is known as *a saddle*.

A physical system that conserves energy corresponds to a dynamical system that preserves volumes in phase space, whereas a physical system that loses energy in time corresponds to a dynamical system that contracts volumes in phase space. Any dynamic system, in which state space volumes are asymptotically invariant under the dynamics, is *conservative*; any dynamic system with shrinking phase volumes is a *dissipative* one.

Roughly speaking, *attractor* of a dissipative systems is an invariant set, any trajectory that starts in stays in. Besides, all neighbouring trajectories converge to this invariant set. The *basin of attraction of an attractor* is the set of points that are driven

towards that specific attractor. Stable equilibrium points and periodic orbits can be given as examples.

*Lyapunov exponents* are generalizations of the eigen-values at equilibrium points. They are used to determine the stability of any type of steady-state behaviour. Trajectories of an $N$-dimensional state space have $N$ Lyapunov exponents, which is often called the *Lyapunov spectrum*. The qualitative features of the asymptotic local stability properties can be summarised by the sign of each Lyapunov exponent. A plus signed Lyapunov exponent indicates an unstable direction, and minus signed Lyapunov exponent indicates a stable direction.

In systems, which have at least one positive Lyapunov exponent, nearby trajectories diverge exponentially fast. This property gives rise to *a sensitive dependence on initial conditions* and is considered as an indicator of *chaotic dynamics*. Due to this very property and the fact that it is impossible to measure the initial states of any real system with infinite accuracy, any initial measurement error in a chaotic system will grow exponentially fast and will eventually reach a magnitude that makes the prediction impossible. Hence, chaotic systems, although predictable in short-term, exhibit *long-term unpredictability*.

Dissipative chaotic systems, as any other dissipative systems, have attractors. These attractors, which are known as *strange attractors*, have *fractal geometry* [1]. Embedded into the strange attractor of a dissipative chaotic system there are many (usually infinitely many) unstable equilibrium points and unstable periodic orbits. These equilibrium points and periodic orbits are of saddle type because repellers cannot be part of an attractor.

Analysing the data taken from a discrete-time system (map) one can easily detect equilibrium points and periodic orbits. Also the equilibrium points of a continuous-time system (flow) can be identified analysing the data. However, the identification of the periodic orbits of a flow from system data requires some auxiliary method, referred to as the *Poincaré Section* method.

This method is applied to an N-dimensional flow by specifying an (N-1)-dimensional hyper-surface together with a reference direction. When a system trajectory pierces the surface in the reference direction, its position on the surface is recorded. The sequence of such piercings constitute a discrete-time process, hence can be described as a map, namely a *Poicaré map*. The equilibrium points and periodic points of this map correspond to periodic orbits of the original flow as illustrated in Figure 1.1. Furthermore, the stability of the equilibrium points or periodic points of the Poincaré map is the same as the stability of the corresponding periodic orbit of the flow.

It should be noted that, in order to be able to detect a periodic orbit of a flow, the Poincaré surface must cut the orbit.

It is clear that, in the process of reducing the flow to a map, the time related knowledge is lost. Indeed, trajectories consume very different lengths of time between successive crossings of the Poincaré surface but they all correspond to one iteration of the map.

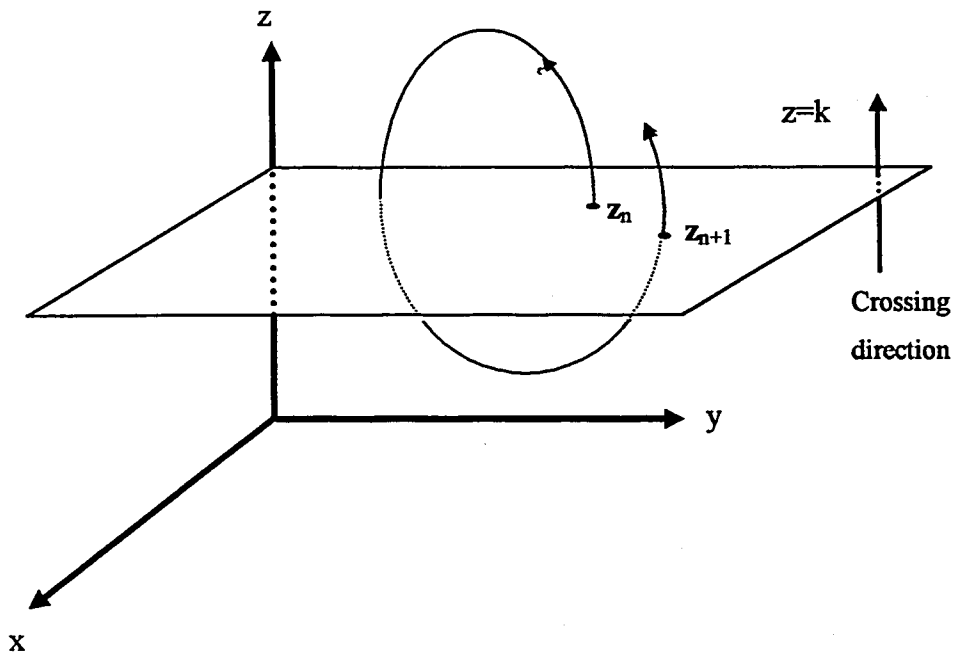



FIGURE 1.1. Poincaré surface of section for a 3D flow

In this example, consider a three dimensional continuous time dynamical system represented by the equation,

$$\dot{\underline{x}} = \underline{F}(\underline{x}) \tag{1.5}$$

where $\underline{x}$ is a state vector as $\underline{x}=[x\ y\ z]^T$ and $\underline{z_n}$ (n=0, ..., number of crossings) is a two dimensional state vector in which successive crossing points stored. Thus, two dimensional Poincaré map corresponding to the system (1.5) can be given as follows,

$$\underline{z}_{n+1} = \underline{G}(\underline{z}_n) \tag{1.6}$$

In most of the cases, it is not possible to write down the map $\underline{G}$ in a closed form.

## 1.2. Some Examples of Chaotic Systems

### 1.2.1. The Lorenz System

Among the first to observe and describe chaotic behaviour in a simple differential equation was Lorenz E. N. His pioneering study describes the numerically observed behaviour of solutions of a system of three first-order ordinary differential equations with simple non-linearities,

$$\dot{x} = -\sigma(x - y) \tag{1.7.a}$$

$$\dot{y} = \rho x - y - xz \tag{1.7.b}$$

$$\dot{z} = xy - \beta z \tag{1.7.c}$$

where $\sigma$, $\rho$ and $\beta$ are system parameters. The Lorenz system exhibits chaotic behaviour for the common known parameter values $\sigma$=10, $\rho$=28 and $\beta$=8/3. Figure 1.2 shows a numerical solution of the Lorenz system for the initial conditions,

$$[x(t_0) \quad y(t_0) \quad z(t_0)] = [0.001 \quad 0.001 \quad 0.001] \tag{1.8}$$

where $t_0$ is the starting time in seconds, while Figure 1.2. shows the states of the system as a function of time. The result has been obtained with 0.001 msec of stepsize.

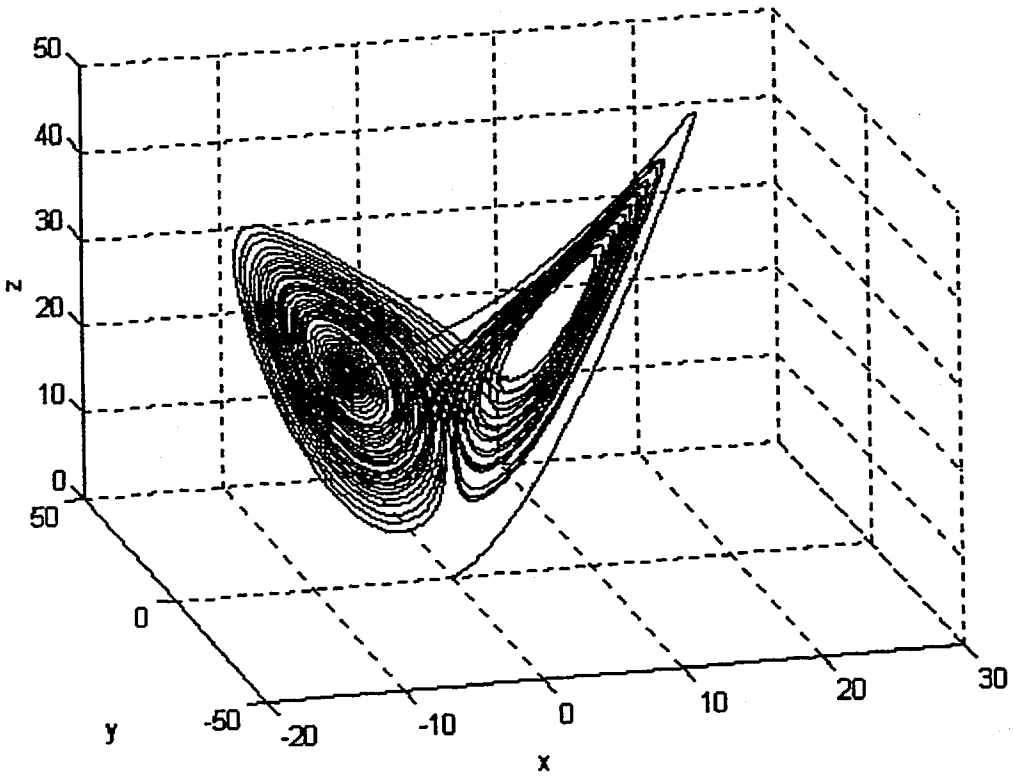



FIGURE 1.2. The strange attractor of the Lorenz system
$\sigma$=10, $\rho$=28, $\beta$=8/z3

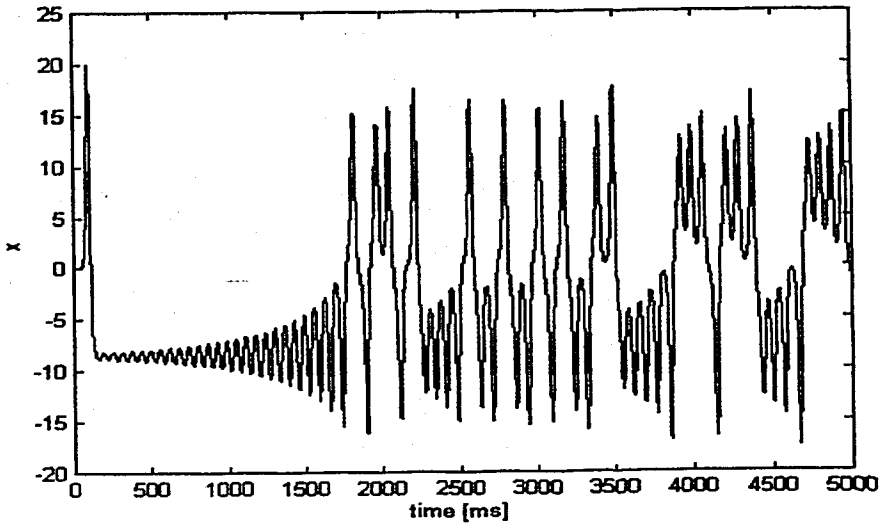



FIGURE 1.3. The state $x$ of the Lorenz system as a function of time

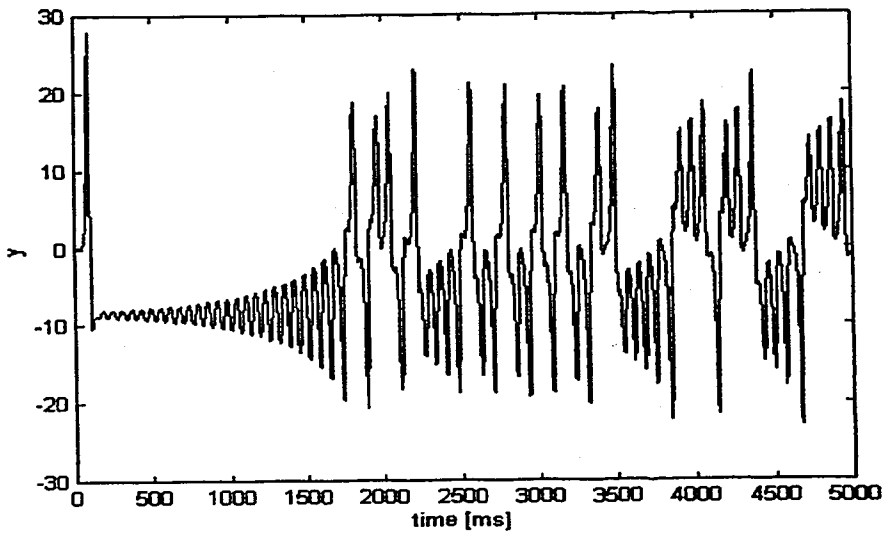



FIGURE 1.4. The state $y$ of the Lorenz system as a function of time

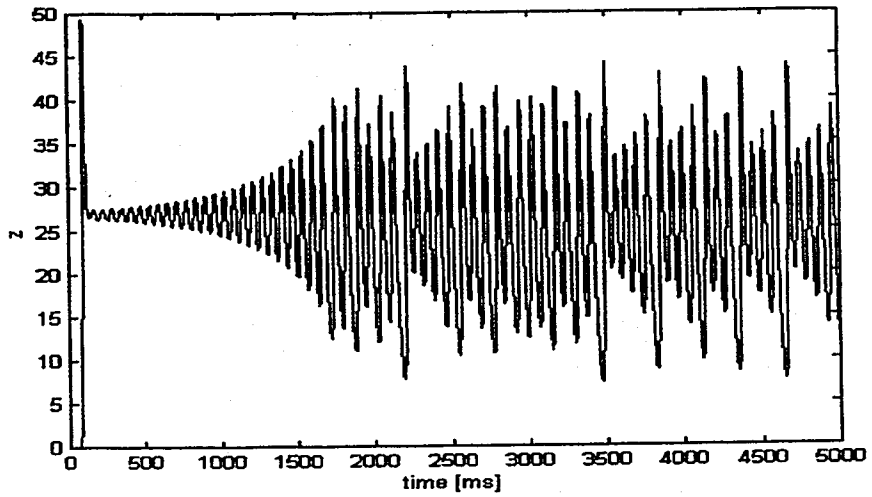



FIGURE 1.5. The state $z$ of the Lorenz system as a function of time

### 1.2.2. The Rossler System

Dynamics of the Rossler system has only a single nonlinear term, and is even simpler than the Lorenz system, which has two nonlinearities, it generates highly nonlinear behaviour governed by the equations,

$$\dot{x} = -y - z \tag{1.9.a}$$

$$\dot{y} = x + ay \tag{1.9.b}$$

$$\dot{z} = b + z(x - c) \tag{1.9.c}$$

where $a$, $b$ and $c$ are the system parameters. The set of nominal parameter values $a$=0.15, $b$=0.20 and $c$=10.00 give chaotic behaviour. Figure 1.6 shows a numerical solution of the Rossler system for the initial conditions,

$$[x(t_0) \quad y(t_0) \quad z(t_0)] = [0.001 \quad 0.001 \quad 0.001] \tag{1.10}$$

where $t_0$ is the starting time in seconds, while Figure 1.6 shows the states of the system as a function of time. The result has been obtained with a step size of 0.001 msec.

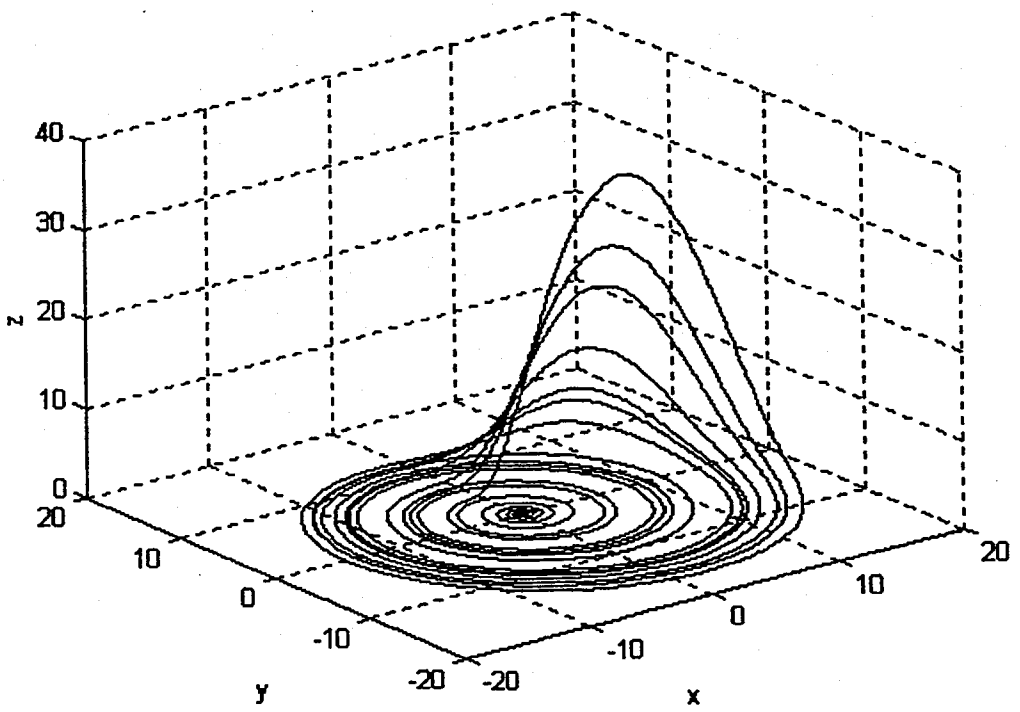



FIGURE 1.6. The strange attractor of the Rossler system
$a$=0.15, $b$=0.20, $c$=10.00

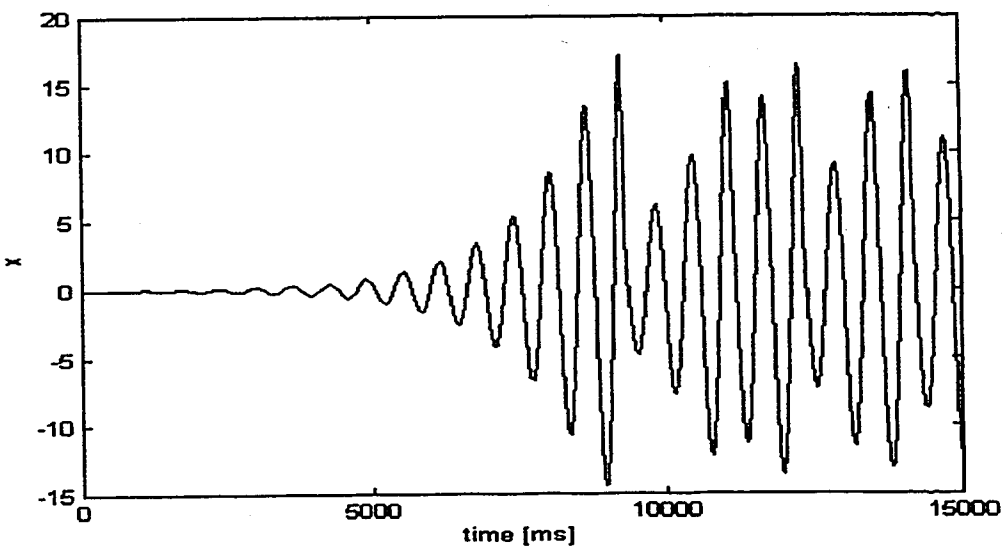



FIGURE 1.7. The state $x$ of the Rossler system as a function of time

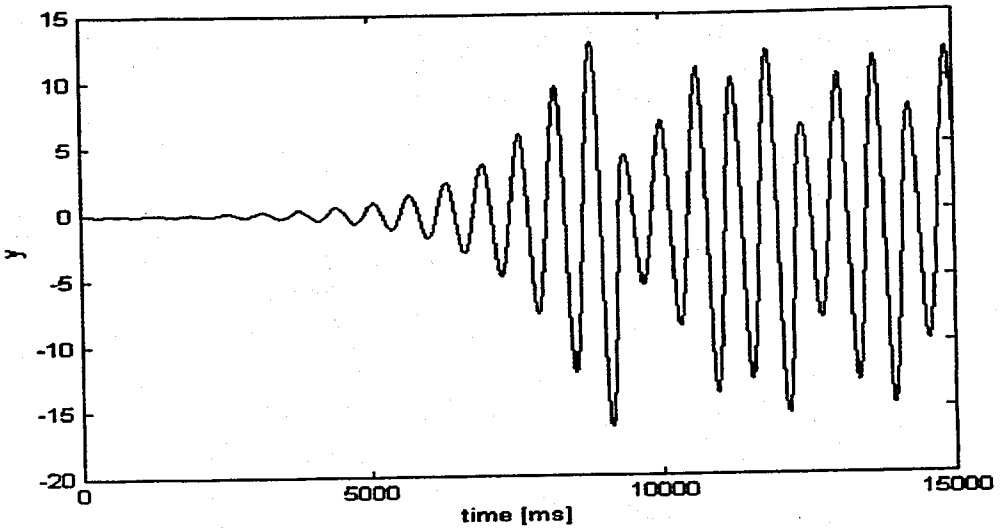



FIGURE 1.8. The state $y$ of the Rossler system as a function of time

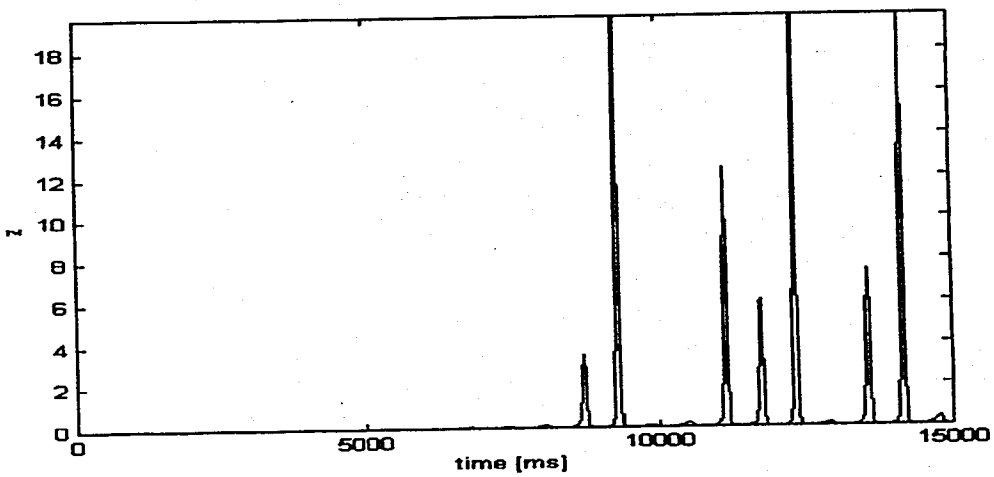



FIGURE 1.9. The state $z$ of the Rossler system as a function of time

### 1.2.3. The Hénon Map

The Hénon map, which was originally proposed by Hénon and Pomeau, is a two dimensional discrete-time non-linear system given by the equations,

$$x_{n+1} = p + 0.3y_n - x_n^2 \tag{1.11.a}$$

$$y_{n+1} = x_n \tag{1.11.b}$$

where $p$ is the adjustable system parameter. Hénon map exhibits chaotic behaviour for some values of $p$. For example for $p=p_{nom}=1.37$, it has a strange attractor and two unstable equilibrium points. Figure 1.10. shows chaotic attractor of the Hénon map for the initial conditions,

$$[x_0 \quad y_0] = [-1.0 \quad -1.7] \tag{1.12}$$

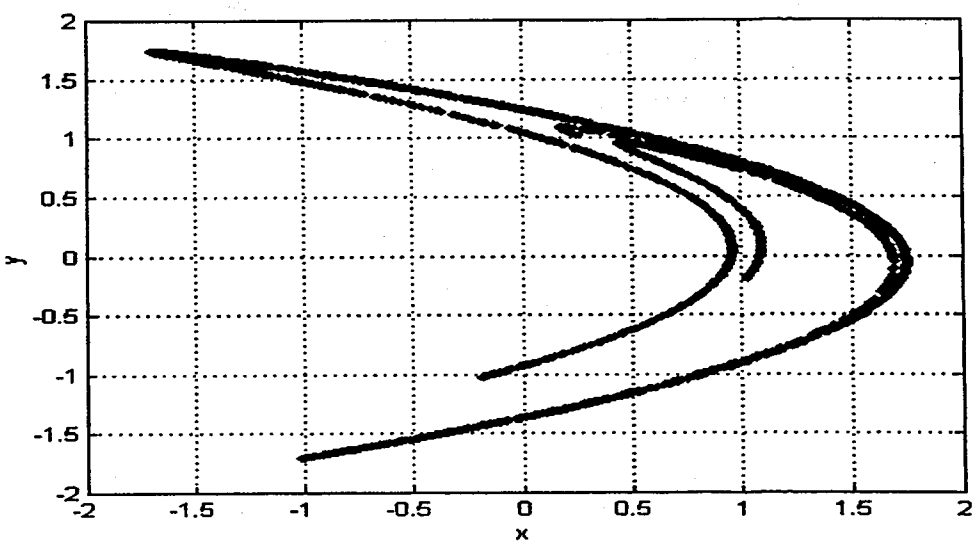



FIGURE 1.10. Chaotic attractor of the Hénon map

### 1.2.4 The Logistic Map

This one-dimensional nonlinear map comes from the population biology and represented by the equation,

$$x_{n+1} = px_n(1-x_n), \quad \text{for} \quad 0 \le x_n \le 1, \quad 0 \le p \le 4 \tag{1.13}$$

where $p$ is the system parameter. The Logistic map exhibits chaotic behaviour for the parameter range given in the equation above. For example for $p=p_{nom}=3.9$, and the initial condition $x_0=0.001$, the time evolution of the Logistic map is given on Figure 1.11.

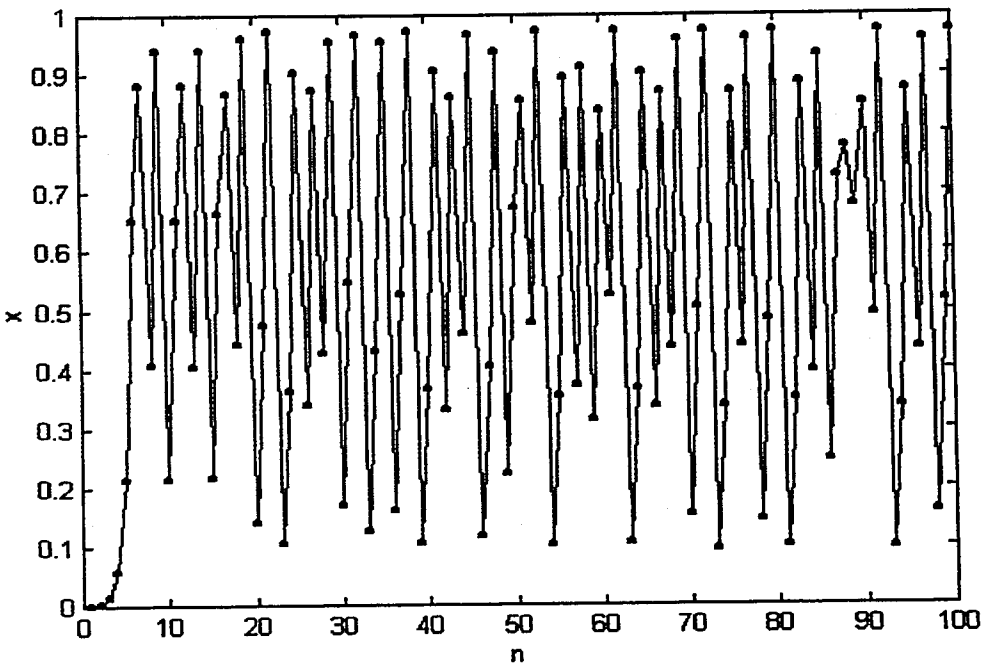



FIGURE 1.11. Chaotic evolution of the Logistic map
(The successive states of the map have been connected with lines in order to improve the visual comprehensibility)

## 1.3. Control of Chaotic Systems

### 1.3.1. General Classification of Chaos Control Approaches

Chaos control is needed in order to analyse and predict the behaviour of chaotic systems. When chaos is a disadvantage to system performance, this usually means reduction of the chaos by stabilising *unstable periodic orbits* (UPO). It is very unlikely that chaotic oscillations are desirable in physical systems and thus there are numerous cases in which chaos is a disadvantage. On the other hand, there are situations in which chaos can be an advantageous property of a non-linear system. An instance of this is when the system is intended for multiple purposes as one could switch among the infinite number of periodic orbits embedded in the chaotic attractor to handle different tasks. Chaos control is a common known concept spanning numerous fields such as medicine, physics, communications, engineering and mathematics. Some of the eminent applications of chaos control will be discussed in detail.

In 1990, Ott, Grebogi and Yorke have proposed the well-known OGY (Ott-Grebogi-Yorke) method [2]. In that paper, they successfully stabilised an unstable fixed point in the Hénon attractor by using small parameter perturbations. The main idea in the OGY method is to select one of the unstable equilibrium points embedded into the system's strange attractor as the goal and stabilise it by small perturbations in a control parameter. In this approach there is no need for the a priori knowledge of the system dynamics. However, a local linear model of the system dynamics in the close neighbourhood of the goal is obtained empirically from the system data. Whenever the system trajectory enters this close neighbourhood a linear feedback control is applied to the system parameter and the originally unstable equilibrium point is stabilised. The major drawback of this method is the long waiting time until the system trajectory enters the close neighbourhood of the selected goal.

The OGY approach has stimulated a great deal of research activity, both in the theoretical and experimental domains. The efficiency of the technique in a periodically forced system has been demonstrated by Ditto et al.[3]. The application of the method to

stabilise periodic orbits in a chaotic diode resonator has been demonstrated by Hunt [4]. The basic OGY method has been applied to many different systems such as chaotic ribbon [5], Nd-doped fiber laser [6], and biological system [7]. Theoretical generalisation of OGY method for higher dimensional systems can be found in Auerbach et al. [8]

Additional to the long-waiting time another disadvantage of the OGY method is the fact that small amounts of noise may cause occasional large departures from the desired trajectory. To increase the effectiveness of the OGY method an initial targeting procedure, which allows the steering of the system trajectory into the close neighbourhood of the goal, has been introduced by Shinbrot et al. [9]. They have experimentally verified the method for vertically oriented, magnetoelastic ribbon [10].

The OGY method has been extended for various purposes. For instance, in 1992, an extension has been proposed that allows a more general feedback matrix, and has been implemented to higher-dimensional systems [11,12]. Two other useful modifications of the OGY method are suggested in [13]. In the following year, the Recursive Proportional Feedback (RPF) method was proposed for highly dissipative systems in [14], and it was experimentally applied to the electrochemical cell in [15]. Lai et al. extended OGY to Hamiltonian systems which cannot be controlled by the classical OGY approach [16]. In 1995, Barreto and Grebogi extended the OGY method to the multiparameter case [17]. A higher-dimensional chaos control extension of the OGY method was proposed in [18], and verified theoretically and experimentally [19]. In the same year, Rhode et al developed a real-time adaptive chaos control method for high-dimensional systems [20]. In [21], Gluckman et al used the extended linearisation matrix proposed in [22], and experimentally implemented an adaptive technique to stabilise and track unstable periodic orbits based on the use of time series.

A different approach to control chaotic systems was proposed by Pyragas [23]. This method is based on the construction of a special form of control, which does not change the system dynamics, but under certain conditions can stabilise the unstable periodic orbit chosen as the target. For this purpose, two different feedback control methods are proposed. A combination of feedback and periodic external force is used in the first method, but the second method does not require any external source of energy and it is

based on self-controlling delayed feedback. The first method of Pyragas can be considered as the special case of the direct application of classical control methods to chaotic systems. This approach can easily be used in experimental systems [24] and improved on [25].

In recent years, control of higher dimensional chaotic systems has gained more attention, and some methods developed by using Lorenz system [26,27,28]. Wiesel has achieved the control of the Lorenz system by using the Linear Quadratic Regulator (LQR) technique [29] and Yu stabilised the Lorenz system by Variable Structure Control method [30].

The unstable periodic orbits of many chaotic systems can be stabilised by some methods described above, but for these methods to be applicable we need to know the location of the desired periodic orbit and the local linearised dynamics in its close neighbourhood. The local linearised model can be obtained from the experimental data using different types of identification systems. In the original article of OGY least square method has been used to identify the local linear model. Later works, which use neural networks both for identification and control, are especially interesting. In 1995, Konishi and Kokame proposed a control system, which stabilises an unstable fixed point by using a neural controller [31] and in 1997 they have been proposed the improved version of the method [32]. The controller successfully works on one and two-dimensional chaotic systems without any a priori knowledge about the system dynamics. Bakker et al have modelled a pendulum by a Feedforward Neural Network, and have used this model to control the pendulum by the Semicontinuous Chaos Control method [33]. Furthermore they utilised the Neural Networks for both predicting and controlling a chaotic system [34].

### 1.3.2. The OGY Method

Ott, Grebogi and Yorke [2], were the first to demonstrate that some properties of chaotic dynamics can be used to control chaotic systems. The method, which is now frequently referred to as the OGY method, can be classified as a parametric perturbation

method. The OGY method is designed to stabilise the Γ unstable periodic orbit of the chaotic system governed by the equation,

$$\dot{\underline{x}} = \underline{F}(\underline{x}, p) \tag{1.14}$$

where, $p$ is an adjustable control parameter and $\underline{x}$ is a $N$-dimensional state vector. Assume that $\underline{x}^*$ is a fixed point of Γ, a closed orbit can be stabilised by varying $p$ appropriately. The control approach is based on the linearization of (1.15),

$$\left.\frac{\partial \underline{x}^*(p)}{\partial \underline{p}}\right|_{\underline{p}=0} \tag{1.15}$$

in the vicinity of $\underline{x}^*$. This is the result of OGY method, which is a local control scheme, if the trajectories passes close enough to $\underline{x}^*$.

In this case, a continuous time dynamical system is considered and the philosophy that lies behind the theory was explained briefly. In the application of the OGY method, basic assumptions are as follows,

(a) The dynamics of the system can be described by an $N$-dimensional map of the form,

$$\underline{z}_{n+1} = \underline{G}(\underline{z}_n, p) \tag{1.16}$$

This map, in the case of continuous time systems, has to be constructed, e.g. the Poincaré surface of section can be used [2]. Since a priori knowledge about the system dynamics is not required, the OGY method is also applicable in experimental studies.

(b) In (1.16), $p$ is an accessible system parameter, which can be varied in some small neighbourhood of its nominal value $p_{nom}$. For the nominal value of $p$, there is a periodic orbit, which is called target, embedded within the chaotic attractor around which

we would like to stabilise the system. It is assumed that, without loss of generality, $p_{nom}$ is set to zero and $p$ is varied in the interval,

$$[-\delta p_{\max}, \delta p_{\max}] \tag{1.17}$$

where $\delta p_{max} > 0$ is the maximum allowable control for external adjustment.

(c) The location of this orbit varies smoothly by changing the parameter $p$ and the effect of the small variations of $p$ retains small in the local system behaviour.

There are large number of unstable equilibrium points (UEP) and unstable periodic orbits (UPO) that can be extracted from an experimentally determined sequence on a chaotic attractor [2]. These experimentally determined UEP's and UPO's are then used to analyse the desired behaviour and one of them is chosen as the target to which the system is to be driven towards. For the purposes of simplicity, it is supposed that the target is a fixed point of period-1 on the Poincaré surface of section. The case of higher periodic orbits is a trivial extension. It is also assumed that the system considered is three dimensional, so Poincaré surface of section is a two dimensional map.

Let $\lambda_s$ and $\lambda_u$ be stable and unstable eigenvalues of the surface of section map (1.16) at the chosen fixed point or target i.e. $\underline{z}^*$. These eigenvalues can be calculated by using the Jacobian matrix described below.

$$\underline{\underline{A}} = \left.\frac{\partial \underline{G}}{\partial \underline{z}}\right|_{z=z^*} \tag{1.18}$$

It is assumed that the stable and unstable eigenvalues satisfy the following inequalities,

$$|\lambda_u| > 1, \quad |\lambda_s| < 1 \tag{1.19}$$

Local directions of the stable and unstable manifolds can be determined by $e_s$ and $e_u$ at the equilibrium point. Here, $e_s$ is the corresponding eigenvector of $\lambda_s$ and $e_u$ is the corresponding eigenvector of $\lambda_u$.

It is clear that the location of the equilibrium point depends on the parameter $p$, i.e. $z^*(p)$. Let the desired fixed point, or equilibrium point, $z^*$ is equal to zero for the sake of simplicity. If the adjustable control parameter $p$ is changed slightly from $p=p_{nom}=0$ to some other value denoted by $p'$, which lies in the interval expressed in (1.17), the location of the equilibrium point will move slightly from 0 to $z^*(p')$. For small $p'$ the approximation can be given by (1.20).

$$b \equiv \left.\frac{\partial \underline{z}^*(p)}{\partial p}\right|_{p=0} \cong p'^{-1}\,\underline{z}^*(p') \tag{1.20}$$

Thus, in the vicinity of the desired equilibrium point $z^*=0$, it is possible to use linear approximation,

$$\underline{z}_{n+1} - \underline{z}^*(p) \cong \underline{\underline{A}}\left[\underline{z}_n - \underline{z}^*(p)\right] \tag{1.21}$$

where, $\underline{\underline{A}}$ is a 2x2 matrix. If we combine the (1.21) with the following,

$$\underline{z}^*(p) \cong pb \tag{1.22}$$

then (1.21) can be written as follows,

$$z_{n+1} \cong p_n b + \left[\lambda_u e_u f_u + \lambda_s e_s f_s\right]\left[z_n - p_n b\right] \tag{1.23}$$

where, $f_s$ and $f_u$ are the contravariant basis vectors expressed by,

$$f_s . e_u = f_u . e_s = 0 \tag{1.24}$$

$$f_s.e_s = f_u.e_u = 1 \tag{1.25}$$

After each crossing of the Poincaré surface, $p$ is adjusted to a new value $p_n$. Thus the value of $p_n$ depends on the location of the current state denoted by $z_n$. If the system is in the close neighbourhood of the desired fixed point denoted by $\underline{z}^*$, a suitable $p_n$ can be calculated so that the next crossing, which id denoted by $\underline{z}_{n+1}$, remains on the stable manifold of the desired fixed point. Theoretically, $p_n$ should satisfy,

$$f_u.\underline{z}_{n+1} = 0 \tag{1.26}$$

If $z_{n+1}$ remains on the stable manifold of $z^*$ then the parameter perturbations are set to zero, $p_n=0$, and the system tends to approach the equilibrium point geometrically with rate rate $\lambda_s$. Driving the system to its stable manifold requires the following control values for parameter $p$ [2].

$$p_n = \begin{cases} \lambda_u(\lambda_u - 1)^{-1}(z_n f_u)/(bf_u) & ,if \quad p_n \in [-\delta p_{\min}, \delta p_{\max}] \\ 0 & ,otherwise \end{cases} \tag{1.27}$$

Basic principle of the OGY method is schematically explained in Figure 1.2. In Figure 1.2.a, if the $n^{th}$ iterate $z_n$ is near the equilibrium point $z^*(p_{nom})$, then in Figure 1.2.b, parameter is changed to $p_{nom} + \delta p$ to move the equilibrium point so as to, in Figure 1.2.c, force the next iterate $z_{n+1}$ onto the stable manifold of the $z^*(p_{nom})$. Then, the perturbation is turned off [3].

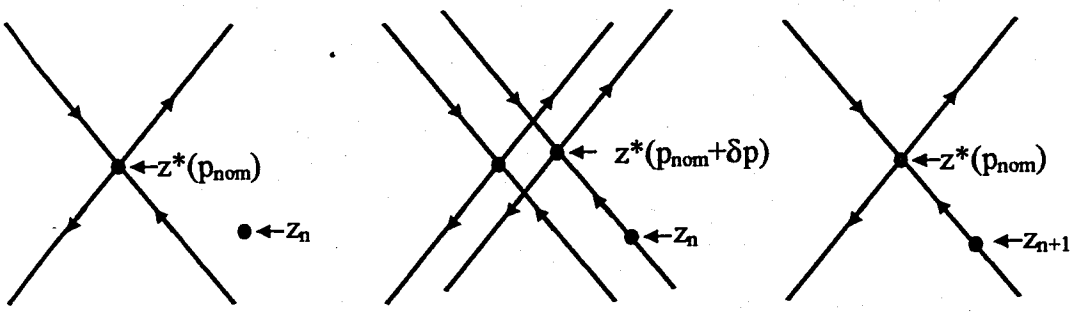



FIGURE 1.14. Sketch of the OGY algorithm;

### 1.3.3. Extension of The OGY Method to Higher Dimensions by Pole Placement Technique

The basic OGY method was extended to higher dimensional systems by constructing a more general feedback matrix [11]. They have demonstrated that chaotic dynamics can be controlled by designing a state feedback controller with respect to an adjustable system parameter. Applications of the method are not limited to cases where, a priori knowledge of the equations governing the dynamics is available. The basic assumptions of the method are as follows:

(a) The location of the desired periodic orbit has to be known.

(b) The linearized dynamics about the periodic orbit has to be known.

(c) The dependence of the location of the periodic orbit on small variation of the control parameter has to be known.

For the purpose of simplicity consider a *N*-dimensional discrete time dynamical system,

$$\underline{z}_{n+1} = \underline{G}(\underline{z}_n, p) \tag{1.28}$$

where, $\underline{z}_n \in R^N$ ,$p \in R$ is the allowable control parameter and $\underline{G}$ is sufficiently smooth function in both variables. It is assumed that $p$ is varied in the interval,

$$[p_{nom} - \delta p_{\max}, p_{nom} + \delta p_{\max}] \tag{1.29}$$

where, $\delta p_{max}$ is the maximum allowable control for external adjustment and $p_{nom}$ is the nominal value of the system parameter. It is also assumed that the system (1.28) has a chaotic attractor for the nominal parameter value, i.e. $p = p_{nom}$.

The objective of the methodology is to change the parameter $p$ among the defined interval (1.29) such that the dynamics of the system approaches to the desired unstable periodic orbit (UPO) or unstable equilibrium point (UEP). A constant state feedback vector is determined by using the linearized dynamics to accomplish the stabilization of the desired UPO or UEP. It is based on the linear system theory that the linearized dynamics of the system is assumed to be stabilizable. For the sake of simplicity, the desired stabilizable target is considered as an unstable fixed point (UPO), $z^*(p)$, i.e. period one orbit. The extension of the method to higher periodic orbits is a trivial work. For the values of $p$, around its nominal value $p_{nom}$, the fixed point of the discrete dynamics, $\underline{z}^*(p_{nom})$ can be approximated by the linear dynamics,

$$\underline{z}_{n+1} - \underline{z}^*(p_{nom}) = \underline{\underline{A}}\left[\underline{z}_n - \underline{z}^*(p_{nom})\right] + \underline{b}(p - p_{nom}) \tag{1.30}$$

where, $\underline{\underline{A}}$ in an $NxN$ Jacobian matrix, $\underline{b}$ is an $N$-dimensional column vector and the partial derivatives are used to evaluate $\underline{\underline{A}}$ and $\underline{b}$ as given in (1.31).

$$\underline{\underline{A}} = D_z G(\underline{z}, p)\Big|_{z^* = z^*(p_{nom})} \tag{1.31.a}$$

$$\underline{b} = D_p G(\underline{z}, p)\Big|_{p = p_{nom}} \tag{1.31.b}$$

It is given in (1.32) that the time dependence of the parameter $p$ can be expressed with a linear dependence to the current state $z_n$.

$$p - p_{nom} = -k^T\left(z_n - z^*(p_{nom})\right) \tag{1.32}$$

In the equation (1.32), $k$ is a $1xN$ state feedback vector, which has to be determined according to the stability condition. The equation (1.33) is obtained by substituting (1.32) into (1.30).

$$\underline{z}_{n+1} - \underline{z}^*(p_{nom}) = \left(\underline{\underline{A}} - \underline{\underline{b}}\underline{\underline{k}}^T\right)\left[\underline{z}_n - \underline{z}^*(p_{nom})\right] \tag{1.33}$$

If the absolute values of the eigenvalues of $(\underline{\underline{A}}\text{-}\underline{\underline{b}}\underline{\underline{k}}^T)$ are smaller than unity, then the matrix $(\underline{\underline{A}}\text{-}\underline{\underline{b}}\underline{\underline{k}}^T)$ is called as *asymptotically stable*. The constant state feedback vector $k$ has to be determined to achieve the stability condition of the matrix $(\underline{\underline{A}}\text{-}\underline{\underline{b}}\underline{\underline{k}}^T)$. The solution of this problem is well known control approach in the classical control theory and is called as *Pole Placement Technique* in the literature [35]. In other words, determine the vector $k^T$ in such a way that the eigenvalues of the matrix $(A\text{-}bk^T)$ have specified values $\{\mu_1, \mu_2, ..., \mu_N\}$, i.e. *regulator poles*. The necessary and sufficient conditions of the pole placement problem are given below:

(a) The pole placement problem has a unique solution if and only if the $NxN$ matrix,

$$\underline{\underline{C}} = \left[\underline{\underline{A}} : \underline{\underline{A}}\underline{\underline{B}} : \underline{\underline{A}}^2\underline{\underline{B}} : \cdots : \underline{\underline{A}}^{N-1}\underline{\underline{B}}\right] \tag{1.34}$$

which is called as *the controllability matrix*, is of rank $N$.

(b) The solution of the Pole Placement problem is given by

$$\underline{k}^T = \left[\alpha_N - a_N \cdots \alpha_1 - a_1\right]\underline{\underline{T}}^{-1} \tag{1.35}$$

where $\underline{\underline{T}}=\underline{\underline{C}}\underline{\underline{W}}$ and

$$\underline{\underline{W}} = \begin{bmatrix} a_{N-1} & a_{N-2} & \dots & a_1 & 1 \\ a_{N-2} & a_{N-3} & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \tag{1.36}$$

$\{a_1, a_2, ..., a_N\}$ are the coefficients of the characteristic polynomial of $A$,

$$\left|s\underline{\underline{I}} - \underline{\underline{A}}\right| = s^N + a_1 s^{N-1} + \dots + a_N \tag{1.37}$$

and $\{\alpha_1, \alpha_2, ..., \alpha_N\}$ are the coefficients of the desired characteristic polynomial of $(\underline{A}\text{-}\underline{b}k^T)$ described in (1.38).

$$\prod_{j=1}^{N} s - \mu_j = s^N + \alpha_1 s^{N-1} + \dots + \alpha_N \tag{1.38}$$

It is clear from the linear system theory that the choice of the state feedback vector $k^T$ can be done in many different ways. As explained above any choice of the regulator poles having an absolute value less than unity achieves stability condition. Note that, the size of the control parameter $p$ is restricted to be in the region (1.29). Hence the combination of (1.29) and (1.32) yields,

$$\left|\underline{k}^T\left(\underline{z}_n - \underline{z}^*(p_{nom})\right)\right| < \delta_{\max} \tag{1.39}$$

The inequality (1.39) defines a slab of width $2\delta/|k^T|$. If the $z_n$ is inside of this slab, control parameter is activated according to (1.32). Otherwise the control parameter is set to its nominal value, i.e. $p=p_{nom}$. These are all summarised in (1.40).

$$p - p_{nom} = -\underline{k}^T\left(\underline{z}_n - \underline{z}^*(p_{nom})\right) u\left(\delta_{\max} - \left|\underline{k}^T\left(\underline{z}_n - \underline{z}^*(p_{nom})\right)\right|\right) \tag{1.40}$$

In (1.40), $u$ is the unit step function defined by (1.41).

$$u(\alpha)=\begin{cases}0, & \alpha \le 0\\ 1, & \alpha > 0\end{cases} \tag{1.41}$$

### 1.3.4. Continuous Control of Chaos by Self-Controlling Feedback

In 1992, a different approach to feedback control was proposed by Pygaras [23]. The method is based on the construction of a special form of a continuous time perturbation. Applied perturbation does not change the form of the desired unstable periodic orbit (UPO) but under certain conditions can stabilize it. Two feedback control loops have been proposed, shown in Figure 1.15 and Figure 1.16, to accomplish this aim. In the first method, a combination of feedback and periodic external force is used, but in the second method any external source of energy is not required since it is based on self-controlling delayed feedback. Neither of the methods requires a priori analytical knowledge of the system dynamics. Especially the first method is suitable for experimental study.

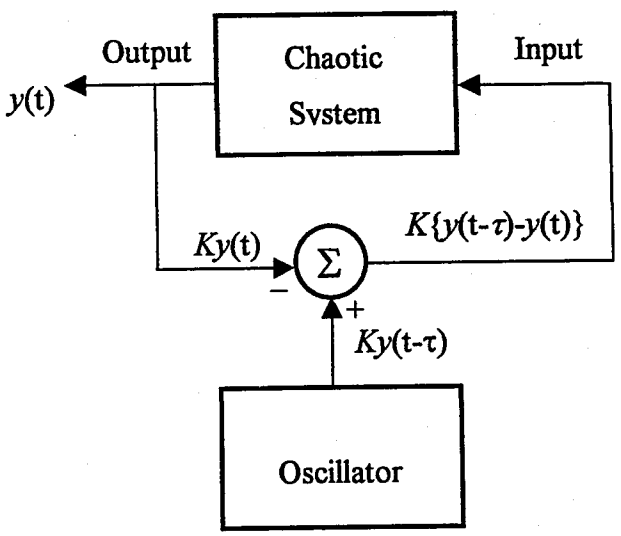



FIGURE 1.15. External force control

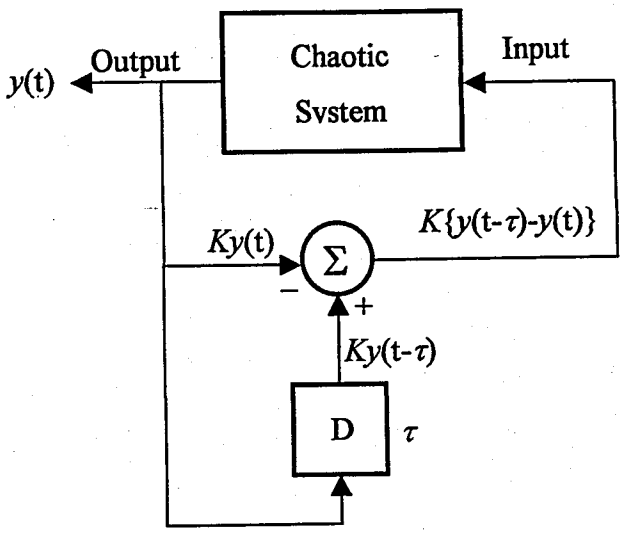



FIGURE 1.16. Delayed feedback control

1.3.4.1. External Force Control. It is assumed that the exact dynamics of the system is unknown and a scalar variable can be measured as system output. It is also supposed that an input variable is available for external force. These are described in (1.41) mathematically.

$$\frac{dy}{dt} = P(y,x) + F(t) \tag{1.41.a}$$

$$\frac{dx}{dt} = Q(y,x) \tag{1.41.b}$$

In (1.41), $y$ is the output variable and the vector $x$ describes the remaining variables of the dynamics, which are not observable. The symbol $F$ in (1.41.a) denotes the external force. If $F$=0 then the system is assumed to have a strange attractor.

It has been proved that a large number of distinct unstable periodic orbits (UPO's) on a chaotic attractor can be obtained from one scalar measurable signal [36]. Applying this method to the system (1.41), the measured output signal $y(t)$ can be determined with different forms $y=y_n(t)$ where $y_n(t+T_n)= y_n(t)$ corresponds to different UPOs embedded on

the strange attractor. The term $T_n$ corresponds the period of the $i^{th}$ UPO. One of the distinct unstable periodic orbits is chosen as a target to be stabilised and a special external force is designed to achieve stability. An appropriate form of the external oscillator is given in (1.42).

$$F(t) = K[y_n(t) - y(t)] \tag{1.42}$$

In (1.42), $K$ is the adjustable weight of the perturbation and can be considered as a negative feedback gain. Note that the perturbation expressed in (1.42) does not change the solution of equation (1.41) corresponding to the UPO $y(t)=y_n(t)$. If the stabilisation is achieved by choosing a suitable negative feedback gain $K$, the output signal stays close to $y_n(t)$ and the magnitude of the external force $F(t)$ decreases to negligibly small values. A small external force is used to stabilise desired UPO as well as in the OGY method. The perturbation in the OGY method is applied only when the state of the system is close to the fixed point. Consequently, in the *External Force Control*, perturbations can be switched on at any moment to stabilise desired periodic orbit. The selection of the feedback gain $K$ has not been proved for any system of interest but for some chaotic systems, verification can be done analytically [23].

To analyse the local stability of a system, the maximal Lyapunov exponent must be calculated, i.e. $\lambda$, of the UPOs using the linearised model of system with respect to small deviations from the corresponding UPOs, i.e. $\lambda(K)$. The negative values of $\lambda(K)$ determine the interval of $K$ corresponding to the stabilised unstable periodic orbit.

1.3.4.2. Delayed Feedback Control: The main disadvantage of the external force control method is the complexity of experimental realisation. Delayed feedback control approach was developed to overcome this difficulty. By substituting the external signal $y_n(t)$ in (1.42) for the delayed output signal $y(t-\tau)$, (1.43) yields the perturbation of the form,

$$F(t) = K[y(t-\tau) - y(t)] \tag{1.43}$$

where, $\tau$ represents a delay time. If the time delay $\tau$ coincides with the period of $i^{th}$ UPO, i.e. $\tau=T_n$, then the perturbation is equal to zero for the solution of the system (1.41) corresponding to desired UPO $y(t)=y_n(t)$.

The delay time $\tau$ and the weight $K$ of the feedback loop should be adjusted experimentally. For some value of $\tau$ and for different initial values, the dispersion of the perturbation $<D^2(t)>$ has to be calculated firstly. The minimum point in the resulted figure gives appropriate time delay$\tau$. This minimum is located at the point of delay time coinciding with the periods of the UPO $\tau=T_n$.

### 1.3.5. Using Chaos to Drive Trajectories to Targets

The OGY control is a local control technique as the state must be close enough to the desired fixed point for the control to be applied to it. The major advantage of the OGY method parametric perturbation strategy among other chaos control techniques is that it can be applied even when a priori analytical model is not known explicitly. Furthermore, the OGY method involves a *waiting time*. This waiting time is the delay before the control can be applied, that is, it is the time taken for the orbit to enter a *small neighbourhood* of the desired unstable periodic orbit. The OGY controller can only be applied in a local region, which will be referred to as the OGY region, about the target and hence a waiting period is introduced. To overcome this problem Shinbrot et al. proposed a targeting technique, which can be applied to reduce the waiting period [9].

The method uses the exponential sensitivity of a chaotic system to tiny perturbations to drive the system to a desired accessible state in a short time. A small, judiciously chosen perturbation to an available system parameter is applied to achieve targeting. The time required to reach an accessible target by applying such a perturbation is formulated successfully. The method proposed is not flexible, since it requires a priori knowledge of the system.

For the purpose of simplicity consider three dimensional continuous time dynamical system given in (1.44).

$$\frac{d\underline{x}}{dt} = \underline{F}(\underline{x}) \tag{1.44}$$

In (1.44), $x$ is a three dimensional state vector. The extension to higher dimensions is a trivial work. Having used a surface of section technique to the system (1.44), the discrete time model can be expressed as a Poincaré map:

$$\underline{z}_{n+1} = G(\underline{z}_n, p) \tag{1.45}$$

where, the map $G$ is invertible function and $p$ is an adjustable system parameter. Suppose that we would like to move from a source point $x_s$ to a small region about a target point $x_t$. Following the trajectory from $x_s$ forward in time, find its first intersection with the surface of section and denote this $z_s$. Similarly following the trajectory through $x_t$ backward in time, the first intersection with the surface of section can be found and denoted by $z_t$. Thus the problem is reduced to that a two dimensional map in which we would like to move from $z_s$ to the vicinity of $z_t$. It is assumed that $p$ is varied within the interval,

$$[p_{nom} - \delta p_{\max}, p_{nom} + \delta p_{\max}] \tag{1.46}$$

where, $\delta p_{max}$ is the maximum allowable control for external adjustment and $p_{nom}$ is the nominal value of the system parameter. If the system considered is ergodic. Then in the absence of perturbations, i.e. $p_n$=$p_{nom}$, the time required to travel from a source point $z_s$, to a small neighbourhood of $\varepsilon_{nomt}$ of linear size $\varepsilon_t$ about a target point $\underline{z}_t$ in the ergodic set is typically expressed as follows:

$$\tau_0 \approx \frac{1}{\mu(\varepsilon_{nomt})} \tag{1.47}$$

where, $\mu$ denotes the natural probability measure of the chaotic set and it is typically scaled with the information dimension $D$. For small $\varepsilon_t$, the time required can be given in (1.48).

$$\tau_0 \approx \frac{1}{\mu(\varepsilon_{nomt})^D} \tag{1.48}$$

It is evident from the equation (1.48) that the time required to reach a desired target increases according to a power law as the diameter of the target region decreases.

If the perturbations are applied to the system, after one iteration of the return map, the change of the state, $\delta z$, relative to the point $G(z_s, p_{nom})$, due to a small perturbation, $\delta_1$, can be given by the Taylor expansion:

$$\underline{\delta z} \cong \left.\frac{\partial G(\underline{z}_s, p)}{\partial p}\right|_{p=p_{nom}} \delta_1 \tag{1.49}$$

If $\delta_1$ varies through the interval $\delta_1 < \delta_*$, equation (1.49) defines a small line segment through the point $G(\underline{z}_s, p_{nom})$. The line segment is denoted by $\delta z'$ and its length is denoted by $\underline{\delta z}$. Since our considerations are limited to chaotic systems, the length of the image of this line segment will grow geometrically with each successive iteration of the map (1.45).

Assume that $n_1$ denotes the number of iterations required for the small line segment to stretch to a length of order one. For small $\underline{\delta z}$, this situation can be observed if the following relation hold:

$$\delta z\, e^{n_1 \lambda_1} \sim 1 \tag{1.50}$$

where, $\lambda_1$ is the positive Lyapunov exponent of the attractor. By defining the equation (1.50), the length of the line segment becomes of order one after about $\tau_1$ iterates, i.e. $\tau_1 \sim n_1$. It is also assumed that the size of the relevant ergodic region to be of the order one, so $\tau_1$ is considered approximately as the number of iterates required for the line segment to span the ergodic region. Similarly, if the region $\varepsilon_{nomt}$ is iterated backward in time, then its preimage spans the ergodic region after a number of preiterates, which is typically of the order of $\tau_2$ as given in (1.51).

$$\tau_2 = |\lambda_2|^{-1} \ln\left(\frac{1}{\varepsilon_t}\right) \tag{1.51}$$

where, $\lambda_2$ is the negative Lyapunov exponent of the map (1.45).

At this point, the following procedure can be applied to drive trajectories to the desired targets. The segment $\delta z'$ is iterated $n_1$ times forward in time using $\alpha_n = \alpha_{nom}$ until its lengths becomes unity. After that, the region $\varepsilon_{nomt}$ is iterated $n_2$ times forward in time until it first intersects the $n_1$ forward iteration of the line segment $\delta z'$. Iterating a point in the middle of this intersection $n_1$ times in the backward direction, we can find a point on the line segment $\delta z$, which is mapped to the target region $\varepsilon_{nomt}$ in $n_1 + n_2$ iterates. By using this, we can determine the required perturbation $\delta_1$ to be applied on the first iterate in the equation (1.49). As a result, the total time to go from $z_s$ to a small neighbourhood of $\varepsilon_{nomt}$ by this method can be given in (1.52).

$$\tau = \tau_1 + \tau_2 = |\lambda_1|^{-1} \ln\left(\frac{1}{\delta z}\right) + |\lambda_2|^{-1} \ln\left(\frac{1}{\varepsilon_t}\right) \tag{1.52}$$

Briefly, it has been demonstrated that it is possible to reach rapidly to a small, accessible target region in a chaotic system by applying small perturbations to an available control parameter.

### 1.3.6. The Extended Control Regions (ECR) Method

The Extended Control Region method to control chaotic systems has been proposed by Iplikci and Denizhan to accomplish local control and targeting tasks [37]. It is assumed that a priori knowledge about the system dynamics is not known, but some of the system parameters are available for external adjustment.

The method considers both targeting and local control issues. Artificial neural networks are used to identify and control for accomplishing the following tasks:

(a) *Targeting Task*: Extension of the control region by taking into account the phase space locations starting from which the system can be brought into the OGY region within K steps with small variations in the control parameters,

(b) *Local Control Task*: Conventional OGY control is applied within the OGY region.

For the sake of simplicity, *N*-dimensional discrete time chaotic systems are considered. If the system dynamics is continuous time, surface of section methods can be used to obtain such a discrete time model.

$$\underline{z}_{n+1} = G(\underline{z}_n, \underline{p}) \tag{1.53}$$

In (1.53), $\underline{z}$ is the $N$x1 state vector and $\underline{p}$ is the $r$x1 adjustable control parameter vector. It is also assumed that the system has an equilibrium point at $z^*$ described by,

$$\underline{z}^* = G(\underline{z}^*, \underline{p}_{nom}) \tag{1.54}$$

where, $\underline{p}_{nom}$ is $r$x1 vector of the nominal parameter values. Allowable control parameter ranges can be defined as in (1.55).

$$\Pi = \{p^i : p^i{}_{nom} - \delta p^i{}_{\max} < p^i < p^i{}_{nom} + \delta p^i{}_{\max}, \quad i = 1,\ldots,r\} \tag{1.55}$$

where, $\delta p^i{}_{max}$ is the maximum allowable parameter change for the $i^{th}$ control parameter.

The main idea behind the ECR method is to identify some regions, ($S_i$, $i$=0, 1, ..., $K$), of the phase space in which neural networks are used either for targeting or local control purposes. The activation region of the controller is extended from $S_0$ (as case of in the OGY control) to the set of all regions, i.e. $S_i$'s ($i$=0, 1, ..., $K$). These regions can be defined as follows:

If a system state $\underline{z}_n$ obeys (1.56) then $\underline{z}_n \in S_0$.

$$\left|\underline{z}_n - \underline{z}^*\right| < \delta, \quad \text{and} \quad \left|G(\underline{z}_n, \underline{p}) - \underline{z}^*\right| < \delta \quad \text{with} \quad \underline{p} \in \Pi \tag{1.56}$$

In other words, starting from any $\underline{z}_n \in S_0$ the system can be kept within the $\delta$ neighbourhood of $\underline{z}^*$ at the next step using a $\underline{p} \in \Pi$. It should be noted that exactly this information gathered from $S_0$ is used by the OGY control method in order to linearise the dynamics of the system.

if $\underline{z}_n \in S_1$ satisfies (1.57),

$$\underline{z}_n \notin S_0, \quad \text{and} \quad G(\underline{z}_n, \underline{p}_n) \in S_0, \quad \underline{p}_n \in \Pi \tag{1.57}$$

Similarly, $\underline{z}_n \in S_2$ if it satisfies (1.58).

$$\underline{z}_n \notin S_0, \quad \text{and} \quad G(G(\underline{z}_n, \underline{p}_n), \underline{p}_{n+1}) \in S_0, \quad \underline{p}_n, \underline{p}_{n+1} \in \Pi \tag{1.58}$$

The definition can be generalised as follows: $\underline{z}_n \in S_i$ if it satisfies (1.59).

$$\underline{z}_n \notin S_0, \quad \text{and} \quad G^{(i)}(\underline{z}_n, \underline{p}_n, \underline{p}_{n+1}, \ldots, \underline{p}_{n+i}) \in S_0, \quad \underline{p}_n, \ldots, \underline{p}_{n+i} \in \Pi \tag{1.59}$$

where,

$$G^{(i)}(\underline{z}_n, \underline{p}_n, \underline{p}_{n+1}, \ldots, \underline{p}_{n+i}) = G(G \ldots (G(\underline{z}_n, \underline{p}_n), \underline{p}_{n+1}), \ldots, \underline{p}_{n+i}) \tag{1.60}$$

In other words, $\underline{z}_n \in S_i$ , $i \neq 0$, can be driven to $S_0$ in $i$ steps by applying control parameters within the allowable range. If the dynamics of the system in the region $S_i$ is successfully modelled, the reaching time can be reduced at an increasing rate as $K$ increases. Thus, in principle the system can be derived from any region $S_i$ into $S_0$ within $i$ steps by applying parameter values within the allowable range; i.e. starting from $S_i$ the system first goes to $S_{i-1}$, then to $S_{i-2}$, $S_{i-3}$ and soon. It should be noted that theoretically a region $S_i$ does not need to be simply connected or connected.

# 2. STATEMENT OF THE PROBLEM

As mentioned in the previous section, the OGY control is concerned with stabilising one of the unstable periodic orbits or equilibrium points, which are embedded into the strange attractor of a chaotic system. These periodic orbits and equilibrium points can be detected and located by analysing the gathered data. Depending on the specific purpose one of them can be chosen as the target behaviour to be stabilised. Thanks to the space-filling property of chaotic dynamics, no matter where it starts from, the phase trajectory of the system will eventually pass through a close neighbourhood of the chosen target. Then a local linear model of the dynamics is used to stabilise the target behaviour by means of small parameter perturbations. The fact that the local model of the dynamics near the target can be obtained from experimental data makes the OGY control a desirable one because it does not require any a priori knowledge about the system dynamics. The basic drawback of this otherwise very advantageous approach is the long waiting time until the phase trajectory enters a close neighbourhood of the target starting from arbitrary initial conditions. In the literature the strategies used in order to make the system enter the close neighbourhood of the target faster are referred to as "*targeting*".

The objective of this thesis was to find a targeting method based on Artificial Neural Networks applicable both to flows and maps. For the local control issues in the OGY region, a non-linear approximation of the controller, namely a neural network model, is used to keep trajectories inside the region. The classical OGY method uses linear approximation for the local system dynamics. In some cases the OGY method is not useful for stabilising some targets, if the linear approximation of the system dynamics in the close neighbourhood of the target is uncontrollable (e.g. Rossler system). In such cases the neural network based local control is advantageous due its inherently non-linear character, which eliminates this deficiency.

In this thesis it is assumed that the target of the flows under consideration is a periodic orbit, although with some extension the method can be made applicable to flows with equilibrium points as their target. For maps no such restriction is necessary.

In this work, as in the original OGY control scheme, it is assumed that no a priori information is available about the system dynamics (accept for its dimension) and that one or several control parameters are available, which can be varied within a narrow interval.

Artificial Neural Networks are used both in order to model the system dynamics partially and in order to control the system. The control process can be divided into two phases: The targeting phase, during which the system is forced to approach the close neighbourhood of the target as soon as possible, and the local control phase, during which a neural network based technique is used to stabilise the target. In that sense the latter phase is equivalent to the OGY method.

In the proposed method all information about the system dynamics is extracted from the experimental data.

# 3. TARGETING VIA REGION SCHEDULING FOR THE CONTROL OF CHAOTIC SYSTEMS

The chaotic systems, which will be considered for the proposed method can be classified as follows according to their type and their target:

(a) An (N+1)-dimensional flow, equation (3.1), with a periodic trajectory as its target.

$$\dot{\underline{x}}(t) = f(\underline{x}(t), \underline{p}) \tag{3.1}$$

where $\underline{x}$ is the $(N+1)$-dimensional state vector and $\underline{p}$ is the $r$-dimensional adjustable control parameter vector.

(b) An N-dimensional map , equation (3.2), with an equilibrium point as its target.

$$\underline{x}_{n+1} = \underline{f}(\underline{x}_n, \underline{p}) \tag{3.2}$$

where $\underline{x}_n$ is the $N$-dimensional state vector and $\underline{p}$ is the $r$-dimensional adjustable control parameter vector.

(c) An N-dimensional map , equation (3.2), with a period-k orbit as its target.

For the purpose of presenting the targeting method all these systems can be represented using the same notation , equation (3.3).

$$\underline{z}_{n+1} = \underline{G}(\underline{z}_n, \underline{p}) \tag{3.3}$$

where $z_n$ is the $N$-dimensional state vector and $p$ is the $r$-dimensional adjustable control parameter vector, Figure 3.1.

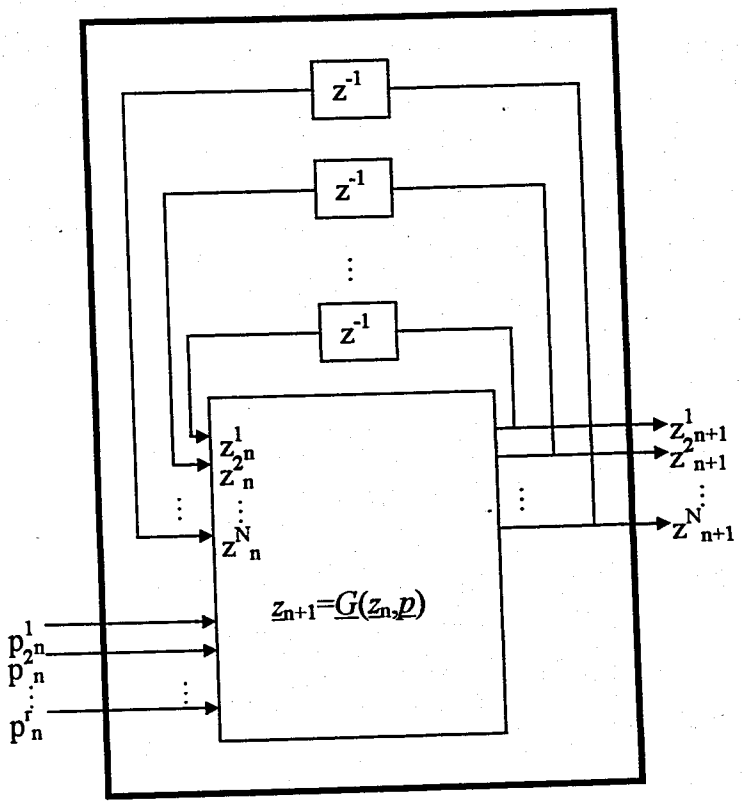



FIGURE 3.1. The model of an *N*-dimensional discrete-time chaotic system with r control parameters

The equilibrium point $\underline{z}^*$ of this system satisfies equation (3.4).

$$\underline{z}^* = \underline{G}\left(\underline{z}^*, \underline{p}_{nom}\right) \tag{3.4}$$

where, $\underline{p}_{nom}$ is the *r*-dimensional nominal control parameter vector.

For case (a), $\underline{G}$ represents the Poincaré map of the flow with an appropriate Poincaré surface and $\underline{z}_n$ is the N-dimensional state vector of the Poincaré map.

For case (b), $\underline{x}_n = \underline{z}_n$ and $\underline{G} = \underline{f}$.

For case (c), $\underline{x}_n = \underline{z}_n$ and $\underline{G} = \underline{f}^{(k)}$.

Using this notation, all cases can be reduced to the problem of targeting for the system represented in equation (3.3), where the target is the equilibrium point $\underline{z}^*$ as defined in equation (3.4).

Allowable control parameter ranges can be defined as given in equation (3.5).

$$\Pi = \left\{ p^i : p^i{}_{nom} - \delta p^i{}_{\max} < p^i < p^i{}_{nom} + \delta p^i{}_{\max}, \quad i = 1,\ldots,r \right\} \tag{3.5}$$

where, $\delta p^i{}_{max}$ is the maximum allowable control for external adjustment for the $i^{th}$ control parameter.

In the proposed method first the strange attractor of the map (3.3) is divided into $K$ non-overlapping regions $R_i$, $i$=1,...,$K$, as can be seen in Figure 3.2. $K$ is chosen heuristically by visual inspection of the data.

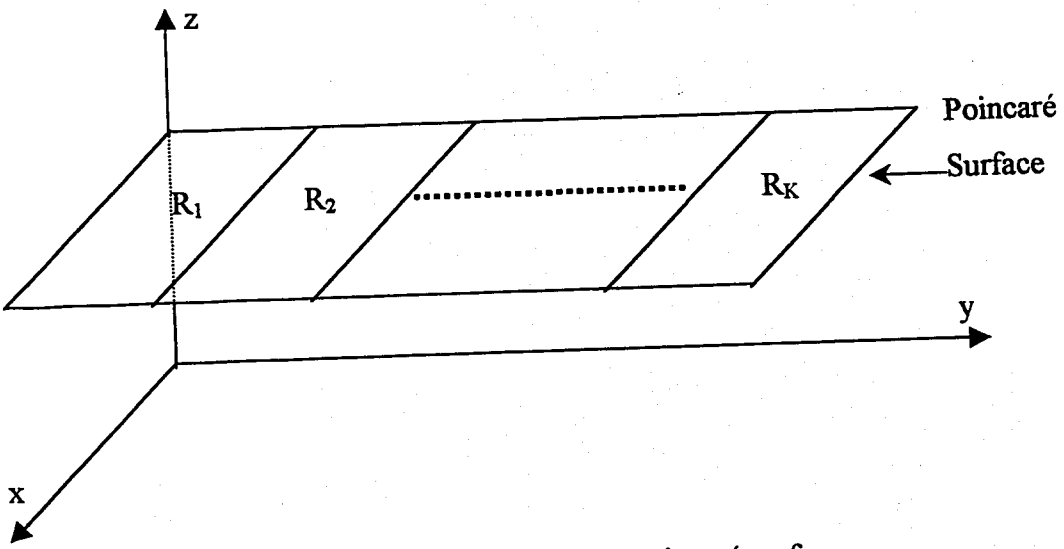



FIGURE 3.2. Control regions on Poincaré surface

Some criteria for this selection will be provided in Section 4. It should be noted that the main idea behind the proposed method is to find a deterministic Finite State Machine (FSM) model for a chaotic attractor and to use it for the purpose of targeting. These regions are going to serve as the states of this FSM.

Next, the experimental data (gathered by varying the control parameter vector) in each region are analysed in order to determine the regions, to which the system can be steered using a $p \in \Pi$ within one step starting from the region under consideration. As a result of this analysis a state transition diagram, Figure 3.3, is obtained, where circles represent states and directed line segments represent transitions between the states of the *finite state machine*.

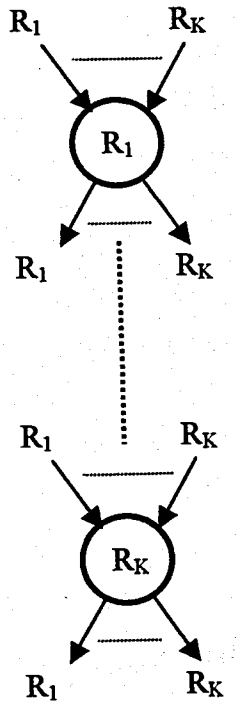



FIGURE 3.3. State transition diagram (FSM model)

Finally, for each initial region the shortest path, i.e. sequence of regions, leading the system to the OGY region is determined together with the corresponding parameter changes. For a system trajectory starting from any given region a sequence of regions leading the system to the OGY region is scheduled. The proposed method is hence referred to as the Region Scheduling (RS) method.

The basic features of the RS control scheme are as follows:

(a) Global control, in contrast with the local in the OGY method.

(b) Guaranteed small control action. One or more of the system parameters, which are available for external adjustment, can be used for the purpose of both targeting and control.

(c) Even if the local linear dynamics of the system under control is uncontrollable such that OGY control becomes impossible, the RS method can stabilise the target.

(d) A priori knowledge of the system dynamics is not required. A set of measured data is used to identify the strange attractor of the chaotic system, to construct a state machine model, and to stabilise a target embedded into the strange attractor.

The realisation of the Region Scheduling method using neural networks will be explained in further detail in Section 4.

To find the possible paths from current region to the region to which the target belongs to, we have used the path planner. The FSM model is considered as a weighted graph and the Dijktra's algorithm is used to find a path between the nodes of the graph constructed. The regions are considered as nodes of the graph and the average time to move between the regions are considered as weights of the graph. An example graph can be seen in Figure 3.4.

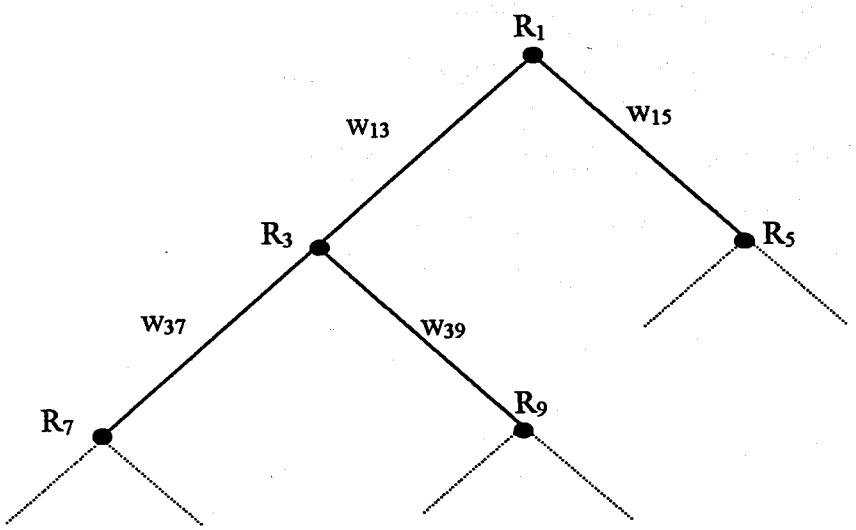



FIGURE 3.4. Weighted graph representation of the FSM

# 4. REALISATION OF THE RS METHOD AND THE LOCAL CONTROL USING ARTIFICIAL NEURAL NETWORKS

The Region Scheduling method can be realised using single hidden layer neural networks. For the selection of the number of nodes in the hidden layer and the training of the neural networks the Radial Basis Functions have been used due to their high approximation capability. In Section 4.1 the Radial Basis Functions are explained.

Section 4.2 is reserved for the explanation of the Path Planning algorithm, which automatically determines the sequence of regions leading an initial state to the close neighbourhood of the target.

Section 4.3 gives the detailed procedure for the neural network realisation of the Region Scheduling method introduced in Section 3.

Finally in Section 4.4 simulation results for some chaotic systems are provided.

## 4.1. Radial Basis Functions

The principle of Radial Basis Functions comes from the theory of functional approximation. Given $N$ pairs,

$$(\vec{x}_i, y_i) \quad (\vec{x} \in R^n, y \in R) \quad , \quad i = 1,\ldots,N \tag{4.1}$$

we want to construct a function $f$ of the form,

$$f(\vec{x}) = \sum_{i=1}^{K} c_i h(|\vec{x} - \vec{t}_i|) \tag{4.2}$$

where, $h$ is the radial basis function and $t_i$ are the $K$ centers, which have to be selected. The model $f$ is a linear combination of radial basis functions. Radial basis functions are a class

of functions, which are ideal for interpolating multivariable data, and which provide exact representations for spherically symmetric data. The coefficients $c_i$ are also unknown and have to be computed. The variables $x_i$ and $t_i$ are elements of an $n$-dimensional vector space. $h$ is applied to the Euclidean distance between each center $t_i$ and the argument vector $x$. Usually, a function $h$ that has its maximum at a distance of zero is used, most often the Gaussian, multiquadratic and inverse multiquadratic function. In this case, values of $x$ which are equal to a center $t$ yield an output value of 1.0 for the function $h$, while the output becomes almost zero for larger distances.

The function $f$ should be an approximation of the given $N$ pairs $(x_i, y_i)$, and should therefore minimise the following error function $H$,

$$H(f) = \sum_{i=1}^{N} (y_i - f(\vec{x}_i))^2 + \lambda \|Pf\|^2 \tag{4.3}$$

The first part of the definition of $H$ (the sum) is the cost, which minimises the total error of the approximation, i.e. it constrains $f$ to approximate the given $N$ points. The second part of $H$ is a stabiliser which forces $f$ to become as smooth as possible, here $\|.\|$ represents the Euclidean norm on $\mathrm{R}^n$. The factor $\lambda$ determines the effect of the stabiliser.

Under certain conditions, it is possible to show that a set of coefficients $c_i$ can be calculated so that $H$ becomes minimal. This calculation depends on the centers $t_i$ which have to be chosen beforehand.

Introducing the following vectors and matrices,

$$\vec{c} = (c_1, \ldots, c_K)^T, \quad \vec{y} = (y_1, \ldots, y_N)^T \tag{4.4}$$

$$G = \begin{pmatrix} h(\|\vec{x}_1 - \vec{t}_1\|) & \cdots & h(\|\vec{x}_1 - \vec{t}_K\|) \\ \vdots & \ddots & \vdots \\ h(\|\vec{x}_N - \vec{t}_1\|) & \cdots & h(\|\vec{x}_N - \vec{t}_K\|) \end{pmatrix}, \quad G_S = \begin{pmatrix} h(\|\vec{t}_1 - \vec{t}_1\|) & \cdots & h(\|\vec{t}_1 - \vec{t}_K\|) \\ \vdots & \ddots & \vdots \\ h(\|\vec{t}_K - \vec{t}_1\|) & \cdots & h(\|\vec{t}_K - \vec{t}_K\|) \end{pmatrix} \tag{4.5}$$

the set of unknown parameters $c_i$ can be calculated by the formula,

$$\vec{c} = \left(G^T G + \lambda G_S\right)^{-1} G^T \vec{y} \tag{4.6}$$

By setting $\lambda$=0, (4.6) becomes identical to the computation of the Moore Penrose inverse matrix, which gives the best solution of an under determined system of linear equations. In this case, the linear system is exactly the one, which follows directly from the conditions of an exact interpolation of the given problem:

$$f(\vec{x}_j) = \sum_{i=1}^{K} c_i h\left(\left|\vec{x}_j - \vec{t}_i\right|\right) \;=\; y_j \quad , \quad j = 1,\ldots,N \tag{4.7}$$

The method of radial basis functions can easily be represented by a three layer feed-forward neural network. The input layer consists of $n$ units, which represent the elements of the vector $x$. The $K$ components of the sum in the definition of $f$ are represented by the units of the hidden layer. The links between input and hidden layer contain the elements of the vectors $t_i$. The hidden units compute the Euclidean distance between the input pattern and the vector, which is represented by the links leading to this unit. The activation of the hidden units is computed by applying the Euclidean distance to the function $h$. Figure 4.1 shows the architecture of the special form of hidden units.

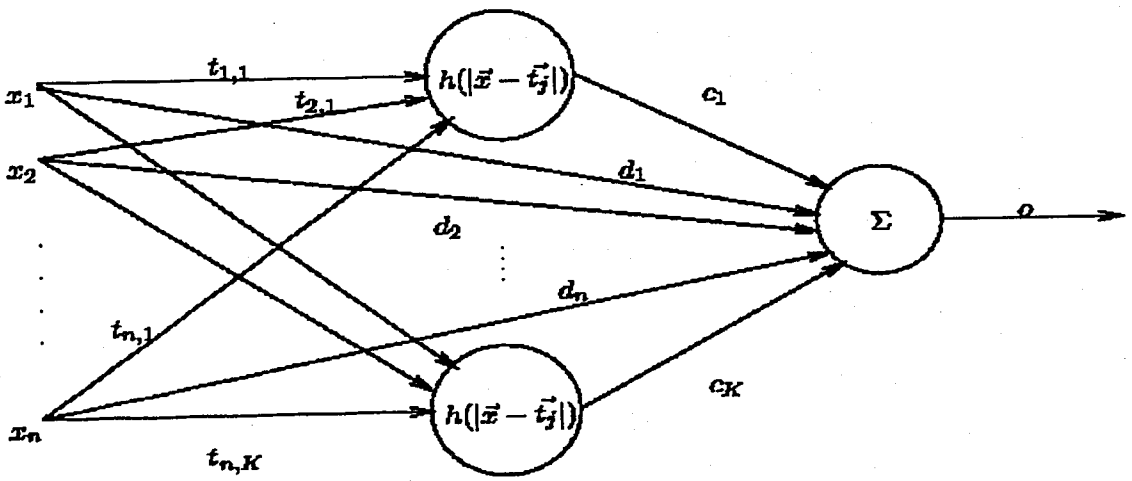



FIGURE 4.1. The architecture of the special form of hidden units

The single output neuron gets its input from all hidden neurons. The links leading to the output neuron hold the coefficients $c_i$. The activation of the output neuron is determined by the weighted sum of its inputs.

The previously described architecture of a neural network, which realises an approximation using radial basis functions, can easily be expanded with some useful features: More than one output neuron is possible which allows the approximation of several functions $f$ around the same set of centers $t_i$. The activation of the output units can be calculated by using a non-linear invertible function $\sigma$ (e.g. sigmoid). The bias of the output neurons and a direct connection between input and hidden layer (shortcut connections) can be used to improve the approximation quality. The bias of the hidden units can be used to modify the characteristics of the function $h$. As a result, a neural network is able to represent the following set of approximations,

$$o_k(\vec{x}) = \sigma\left(\sum_{j=1}^{K} c_{j,k} h\left(\left|\vec{x} - \vec{t}_j\right|, p_j\right) + \sum_{i=1}^{n} d_{i,k} x_i + b_k\right) = \sigma\left(f_k(\vec{x})\right), \quad k = 1,\ldots,m \tag{4.8}$$

This formula describes the behaviour of a fully connected feed-forward network with $n$ input, $K$ hidden and $m$ output neurons. $o_k(x)$ is the activation of output neuron $k$ on the input,

$$\vec{x} = x_1, x_2, \ldots, x_n \tag{4.9}$$

to the input units. The coefficients $c_{j,k}$ represent the links between hidden and output layer. The short-cut connections from input to output are realised by $d_{i,k}$. $b_k$ is the bias of the output units and $p_j$ is the bias of the hidden neurons, which determines the exact characteristics of the function $h$. The activation function of the output neurons is represented by $\sigma$.

The most important advantage of the method of radial basis functions is the possibility of a direct computation of the coefficients $c_{j,k}$ (i.e. the links between hidden and output layer) and the bias $b_k$. This computation requires a suitable choice of centers $t_j$ (i.e.

the links between input and hidden layer). Because of the lack of knowledge about the quality of the $t_j$, it is recommended to append some cycles of network training after the direct computation of the weights. Since the weights of the links leading from the input to the output layer cannot be computed directly, there must be a special training procedure for neural networks that uses radial basis functions. The implemented training procedure tries to minimise the error $E$ by using gradient descent. It is recommended to use different learning rates for different groups of trainable parameters. The following set of formulas contains all information needed by the training procedure,

$$E = \sum_{k=1}^{m}\left(\sum_{i=1}^{N}\left(y_{i,k} - o_k(\vec{x}_i)\right)^2\right) \tag{4.10}$$

$$\Delta\vec{t}_j = -\eta_1 \frac{\partial E}{\partial \vec{t}_j} \tag{4.11.a}$$

$$\Delta p_j = -\eta_2 \frac{\partial E}{\partial p_j} \tag{4.11.b}$$

$$\Delta c_{j,k} = -\eta_3 \frac{\partial E}{\partial c_{j,k}} \tag{4.11.c}$$

$$\Delta d_{i,k} = -\eta_3 \frac{\partial E}{\partial d_{i,k}} \tag{4.11.d}$$

$$\Delta b_k = -\eta_3 \frac{\partial E}{\partial b_k} \tag{4.11.e}$$

It is often helpful to use a momentum term. This term increases the learning rate in smooth error planes and decreases it in rough error planes. The next formula describes the effect of a momentum term on the training of a general parameter $g$ depending on the additional parameter $\mu$. $\Delta g_{t+1}$ is the change of $g$ during the time step $t+1$ while $\Delta g_t$ is the change during time step $t$,

$$\Delta g_{t+1} = -\eta \frac{\partial E}{\partial g} + \mu \Delta g_t \tag{4.12}$$

Another useful improvement of the training procedure is the definition of a maximum allowed error inside the output neurons. This prevents the network from getting over trained, since errors that are smaller than the predefined value are treated as zero. This in turn prevents the corresponding links from being changed.

## 4.2. Path Planning Algorithm

Dijkstra's algorithm solves the shortest path connecting two specified vertices on a weighted, directed graph $G = (V, E)$, $V$ is a set of vertices and $E$ is a set of edges, for the case in which all edge weights are nonnegative.

For each edge $e= (a, b)$ of the graph $G = (V, E)$ a nonnegative real number is assigned as *the weight* of $e$ and denoted by $wt(e)$ or $wt(a, b)$, where $a$ and $b$ are used to define *nodes*. For any $e= (a, b) \in E$ $wt(e)$ represents the length of a total path from $a$ to $b$. If $a, b \in V$ but $(a, b) \notin E$, the weight is defined as $wt(a, b)=\infty$.

Given a weighted graph $G = (V, E)$, for each $e= (a, b) \in E$ it is supposed to find $d(a, b)$, the shortest distance from $a$ to $b$. If no such path exists in $G$ from $a$ to $b$, then the distance can be defined as $d(a, b)=\infty$, and we can also express that for any $a \in V$, $d(a, a)=0$. Thus the distance function $d: V \times V \rightarrow R^+ \cup \{0. \infty\}$.

For any $v_0 \in V$ determine all $v \in V$ such that $d(v_0, v)$ is minimum. This is the objective of the algorithm. To accomplish this objective, the algorithm is designed by E. B. Dijkstra in 1959 [38]. The details and the proof of the algorithm can be found on the cited reference.

To illustrate the application of the algorithm, consider a weighted graph having 10 nodes $\{N_1, ..., N_{10}\}$ as shown in Figure 4.2. Minimum path from the source node $N_1$ to the destination node $N_7$ can be calculated by the Dijktra's algorithm and the result is as follows.

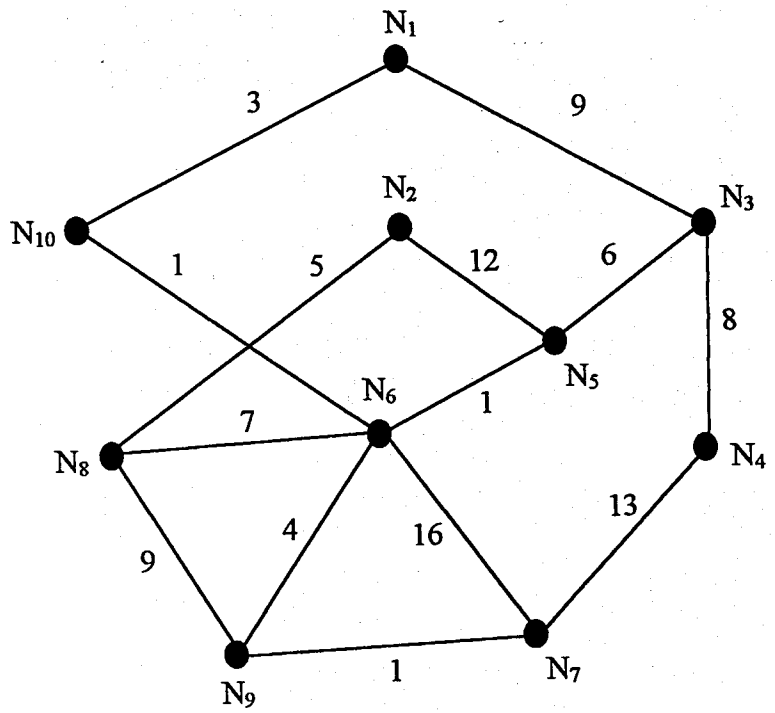



FIGURE 4.2. An example graph
(optimal path:N1, N10, N6, N9, N7 and the total weight: 9)

## 4.3. Off-LineTraining and On-Line Usage of the NN Controllers

### 4.3.1. The Off-Line Training of the Controllers

The first step of the realisation of RS method is to collect sufficient number of data sets by varying the control parameters within the allowable range f randomly. These data sets are stored in the form of $\{z_n, z_{n+1}, p_n\}$. It is assumed that all states of the system are measurable, otherwise other techniques for the reconstruction of the phase space have to be used as explained in [39].

Next, the phase space covered by the data is partitioned into regions $R_i$, $i = 1,...,K$. The data is analysed to determine $C(R_i, R_j)$, the relative frequency of all $\underline{z}_n \in R_i$ which can be taken to a $\underline{z}_{n+1} \in R_j$ using a $\underline{p}_n \in$ f. A weighted graph representing possible transitions between the regions is constructed, where $C(R_i, R_j)$'s are used as the weighing coeficients corresponding to the edge connecting $R_i$ to $R_j$. Dijktra's algorithm is applied to this graph in order to construct a look-up table, which shows the shortest path (i.e. sequence of regions) leading from any region to the target region.

On the other hand, the data sets $\{z_n, z_{n+1}, p_n\}$ with $\underline{z}_n \in R_i$ are used to train the $i^{th}$ neural network in an off-line supervised manner. Hence, $K$ neural networks corresponding to the regions $R_i$ ($i$=1,...,$K$) are trained to realise the non-linear map given in (4.13). Figure 4.3 illustrates the training scheme.

$$\underline{p}_n = \underline{G}_i(\underline{z}_{n+1}, \underline{z}_n) \qquad \text{for } z_n \in R_i, \tag{4.13}$$

### 4.3.2. The On-Line Usage of the Controller

In the on-line control mode first the region of the current state the current state $\underline{z}_n$ is determined. The shortest path leading to the target region is found from look-up table.

The current state $\underline{z}_n \in R_i$ and an arbitrary point within the next region $R_j$ scheduled by the path planner are fed into the neural network corresponding to $R_i$. The output of this neural network is the parameter vector $\underline{p}_n$ necessary to realise this $R_i$-$R_j$ transition. If $\underline{p}_n$ provided by the neural network is outside the allowable range $\Pi$ the $p_n$ is set equal to $p_{nom}$ (equation 4.14), i.e. the system is allowed to evolve in an uncontrolled manner according to its own dynamics.

$$p_n = \begin{cases} p_{NN} & \text{if } p_n \in \Pi \\ p_{nom} & \text{if } p_n \notin \Pi \end{cases} \tag{4.14}$$

Usually $p_n \notin \Pi$ is obtained due to the insufficiency of the neural networks to model the regions, which actually have a fractal geometry.

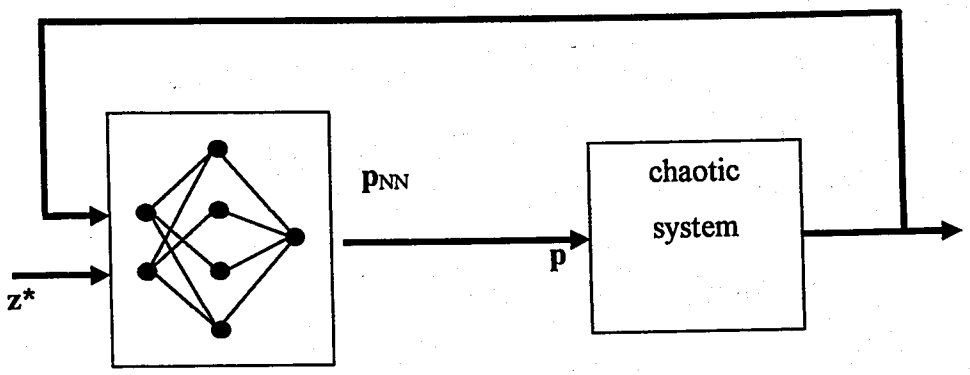



FIGURE 4.3. Simple system architecture

## 4.4. Simulation Results

In this study, simulations are performed on the Lorenz system, the Rossler system, the Hénon map and the Logistic map. The basic OGY controller [2] and the Neural Network Based ECR controller [37] are compared with the RS method with respect to the following items:

(a) the amount of data required to find a model dynamics and a controller,

(b) the amount of time required to find a model dynamics and a controller,

(c) the average tracking error,

(d) the average control effort,

(e) the average time needed to reach a close neighbourhood of the target.

### 4.4.1. The Lorenz System

It is assumed that all states of the Lorenz system are available for measurement. Figure 4.4 illustrates the control architecture for the Lorenz system. The complete set of measured states constitutes the input of the Poincaré section module. In this simulation, the Poincaré surface is chosen at y = 8.47, and the crossing direction is chosen as dy/dt < 0. Three control parameters are assumed to be available for external adjustment within the allowable range. Thus, a multi-parameter control has been employed. In Figure 4.4, ($z_{1d}$, $z_{2d}$) is the equilibrium point vector of the Poincaré map or the desired vector on any region.

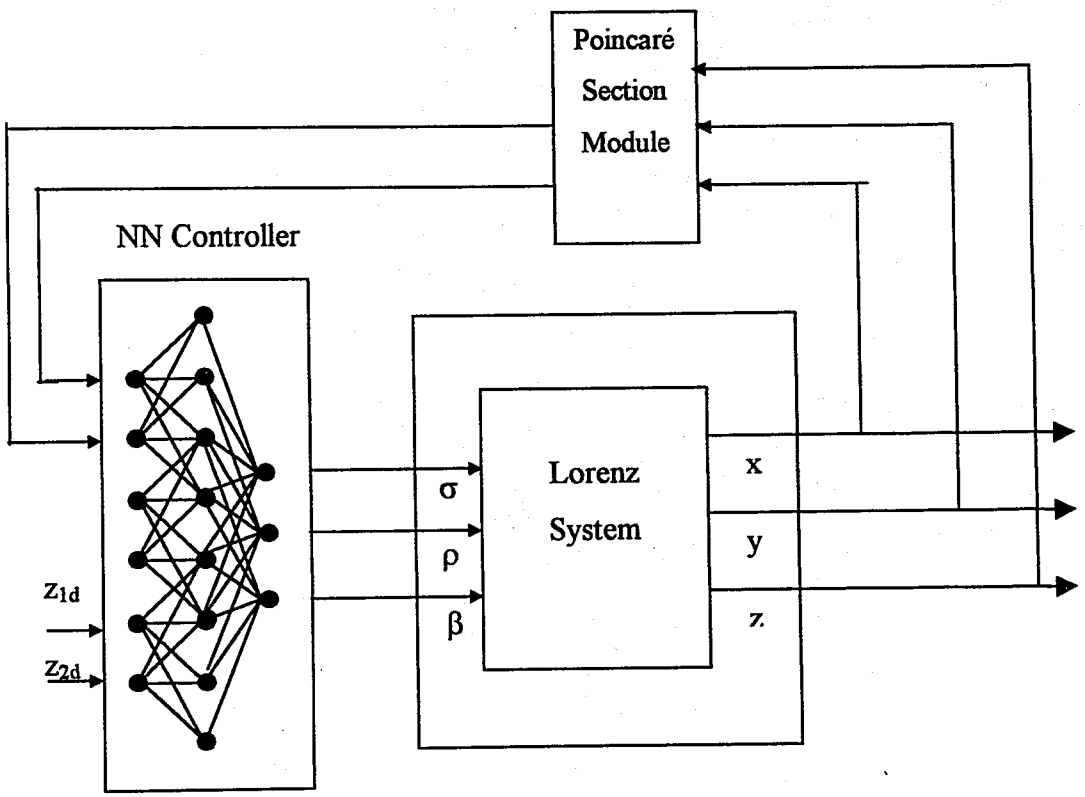



FIGURE 4.4. Control system for the Lorenz equations

Stabilisation of the unstable periodic orbit of the Lorenz system is achieved by using the OGY method, the Neural Network Based ECR method and the RS method. For these controllers, the numerical results are given in the Table 4.1. The outputs of the Lorenz system controlled by the controllers can be seen from Figure 4.5 to Figure 4.7. In

Figure 4.6, the one should note that control parameter changes are equal to zero for some time at the beginning, since the neural network based controllers can not evaluate suitable control parameters in the allowable parameter range Π. Thus, the nominal parameters are used to keep the exact dynamics of the system under control whenever they are not in the allowable range.

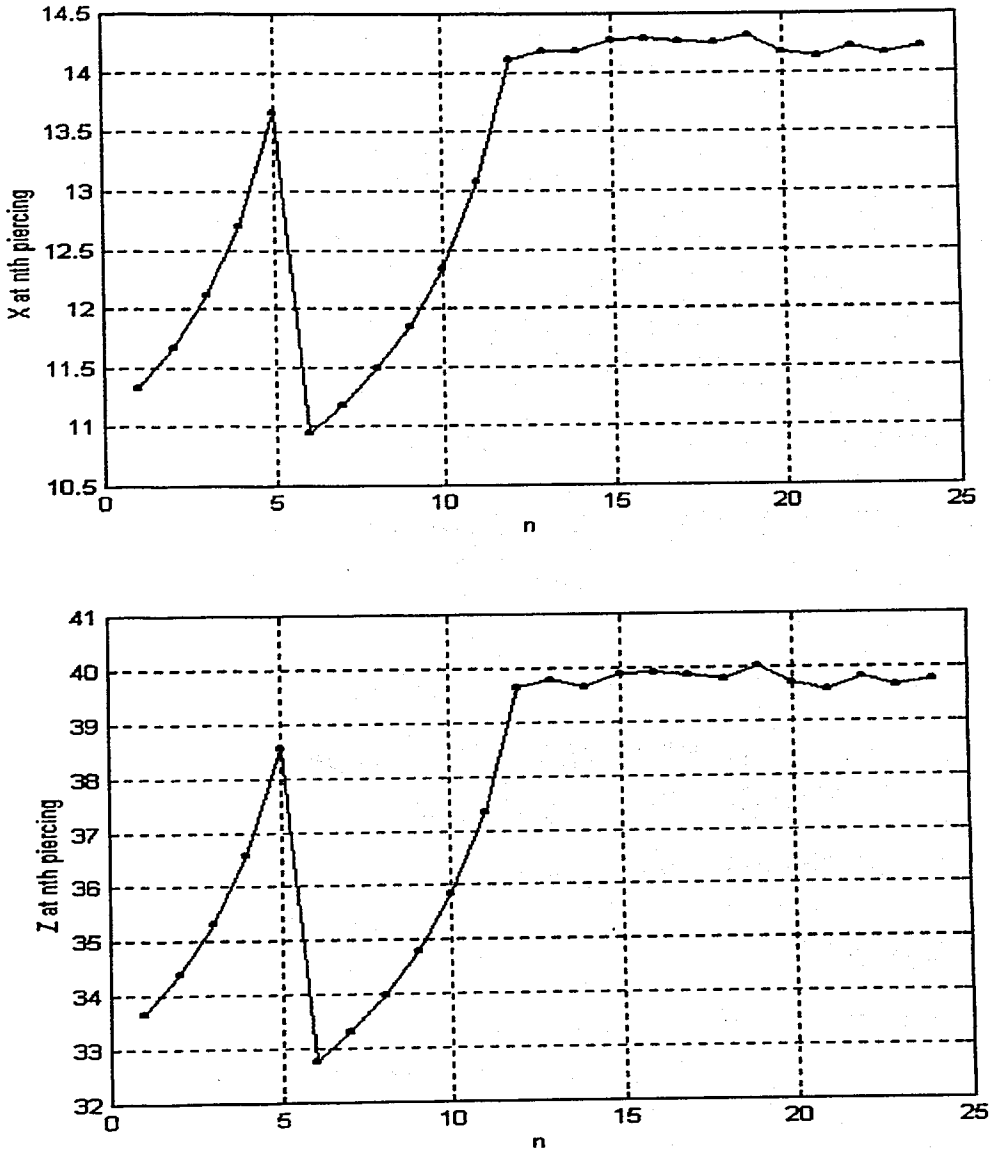



FIGURE 4.5. States of the Lorenz system controlled by the RS method

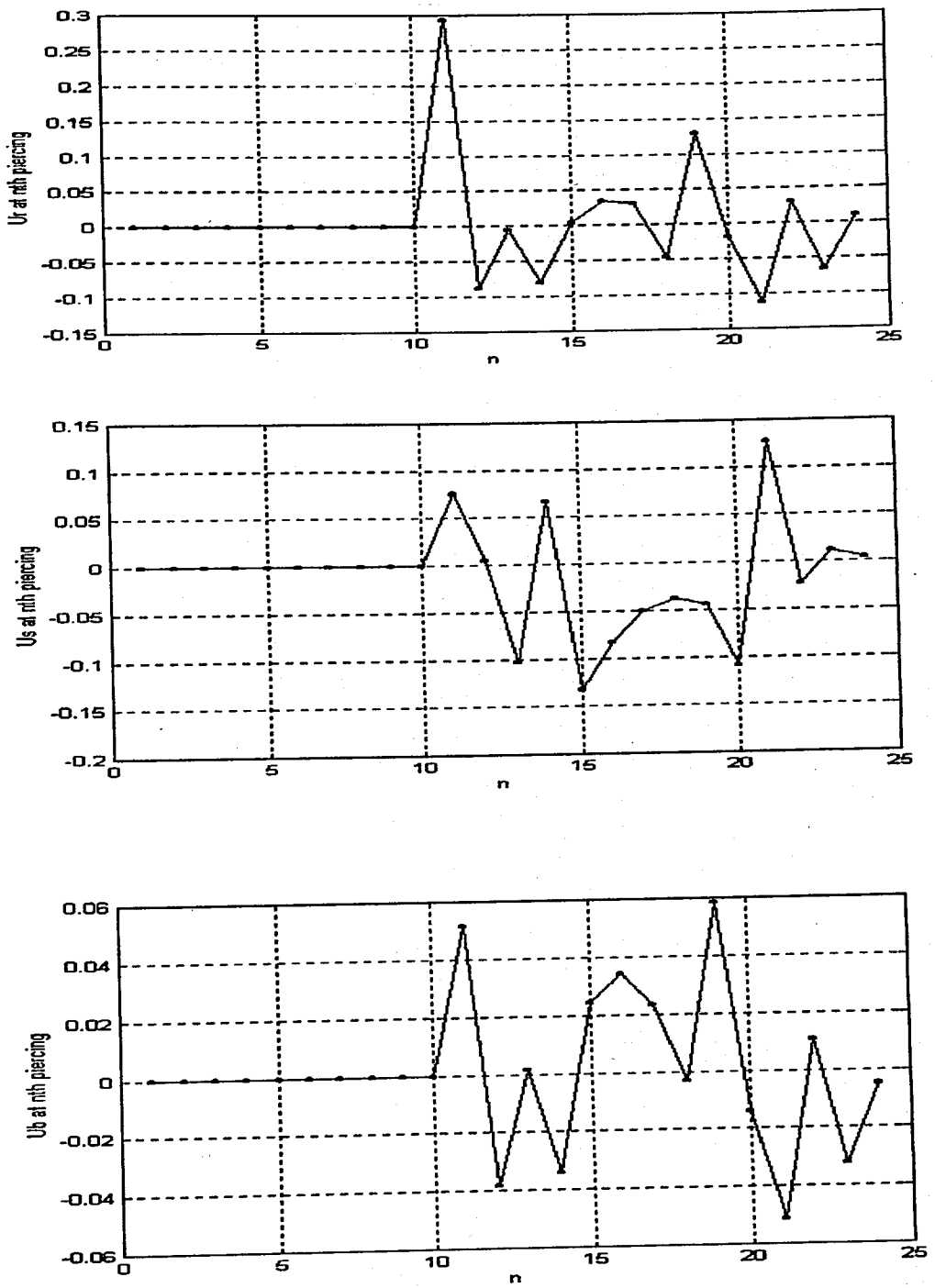



FIGURE 4.6. Change in control parameters about their nominal values for the Lorenz system controlled by the RS method

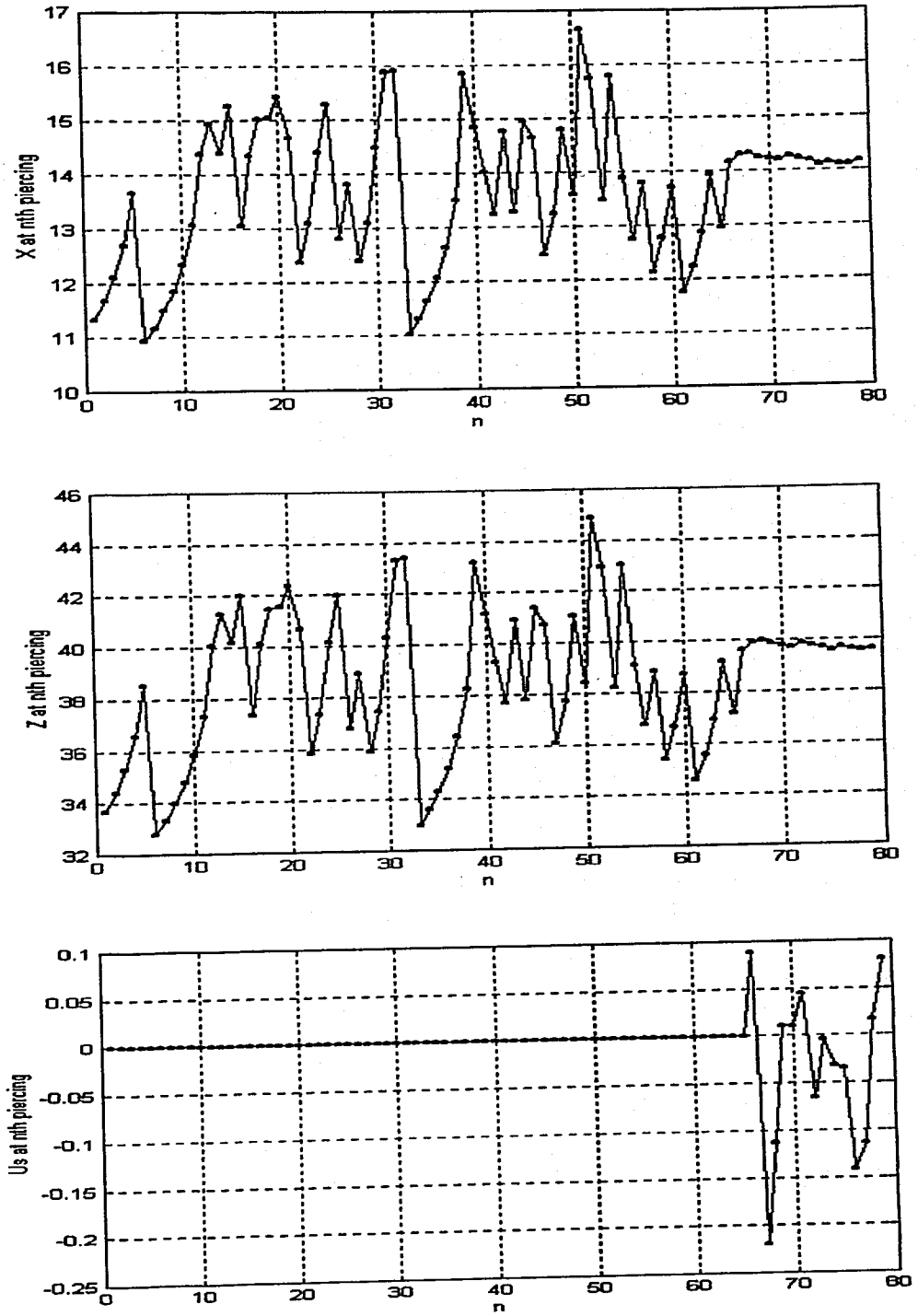



FIGURE 4.7. Change in control parameters about their nominal values and the states of the Lorenz system controlled by the OGY method

TABLE 4.1. Comparison of the controllers for the Lorenz system

| | The OGY Method | The NN Based ECR | The RS Method |
|---|---|---|---|
| Number of control regions | 0 | 1 | 10 |
| Radius of the target region ($\delta$) | 0.3 | 0.3 | 0.3 |
| Max. Allowed par. change ($\delta\sigma_{max}$) | 0.3 | 0.3 | 0.3 |
| Max. Allowed par. change ($\delta\rho_{max}$) | 0.84 | 0.84 | 0.84 |
| Max. Allowed par. change ($\delta\beta_{max}$) | 0.08 | 0.08 | 0.08 |
| Training set (patterns) | 119 | 238 | 238 |
| Training time (CPU time) | 62470 | 96940000 | 9736570000 |
| Average deviation from target | 0.074 | 0.088 | 0.0628 |
| Average deviation from $\sigma_{nom}$ | 0.00590000 | 0.13470000 | 0.0353 |
| Average deviation from $\rho_{nom}$ | 0.04154000 | 0.38720000 | 0.0420 |
| Average deviation from $\beta_{nom}$ | 0.01611000 | 0.03189000 | 0.0133 |
| Avr. reaching time(piercing) | 15.09 | 6.64 | 3.7725 |

### 4.4.2. The Rossler System

It is assumed that all states of the Rossler system are available for measurement. Figure 4.5 illustrates the control loop for the Rossler system. The complete set of measured states constitutes the input of the Poincaré surface of section module. In this simulation, the Poincaré surface is chosen at y = -8.0, and the crossing direction is chosen as $dy/dt < 0$. Three control parameters are assumed to be available for external adjustment within the allowable range. Thus, multi-parameter control has been employed. In Figure 4.8, ($z_{1d}$, $z_{2d}$) is the equilibrium point vector of the Poincaré map or the desired vector on any region.

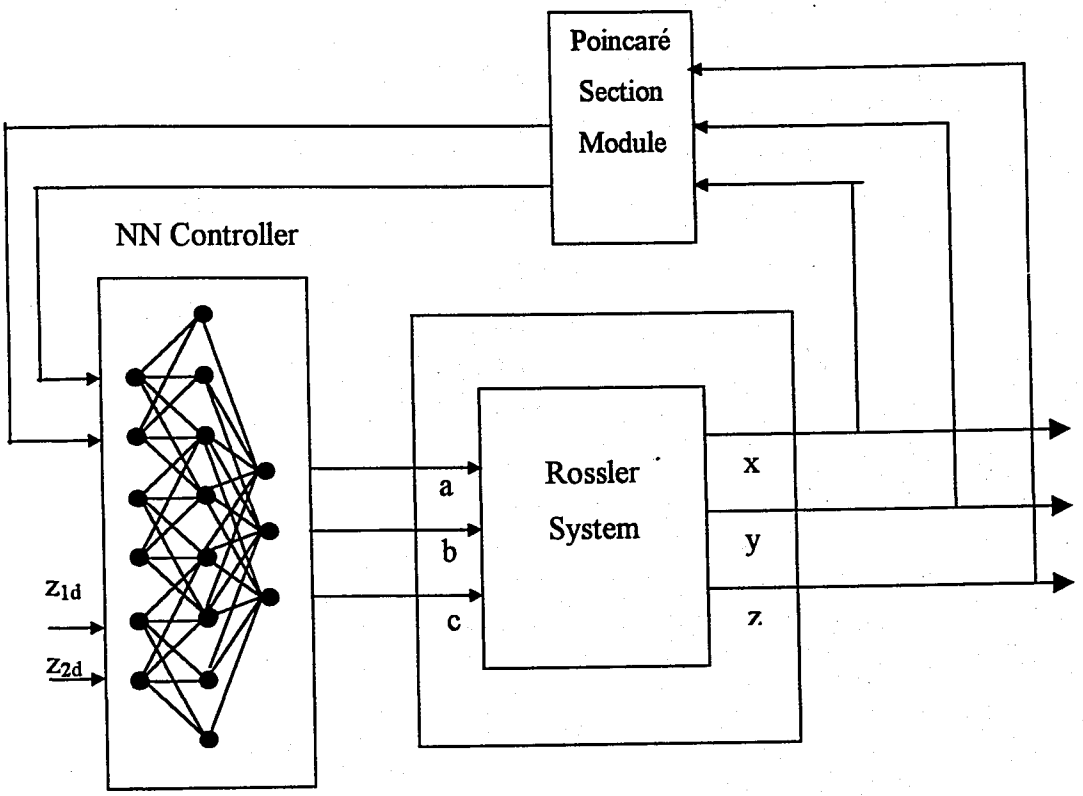
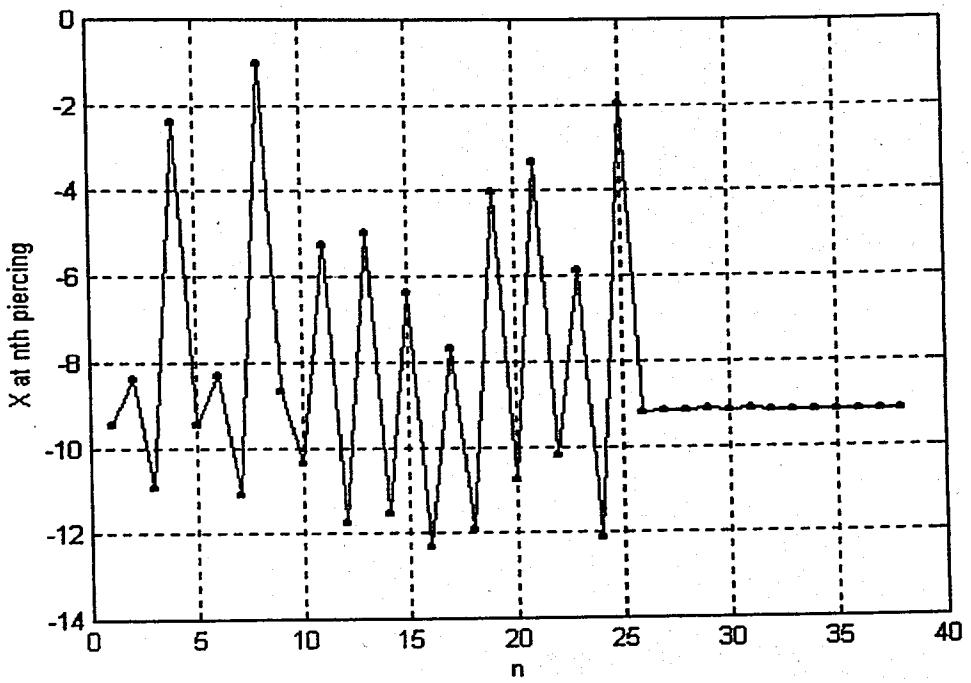



FIGURE 4.8. Control system for the Rossler equations

Stabilisation of the unstable periodic orbit of the Rossler system is achieved by using the OGY method, the Neural Network Based ECR method and the RS method. For these controllers, the numerical results are given in the Table 4.2. The outputs of the Rossler system controlled by the controllers can be seen from Figure 4.9 to Figure 4.11.



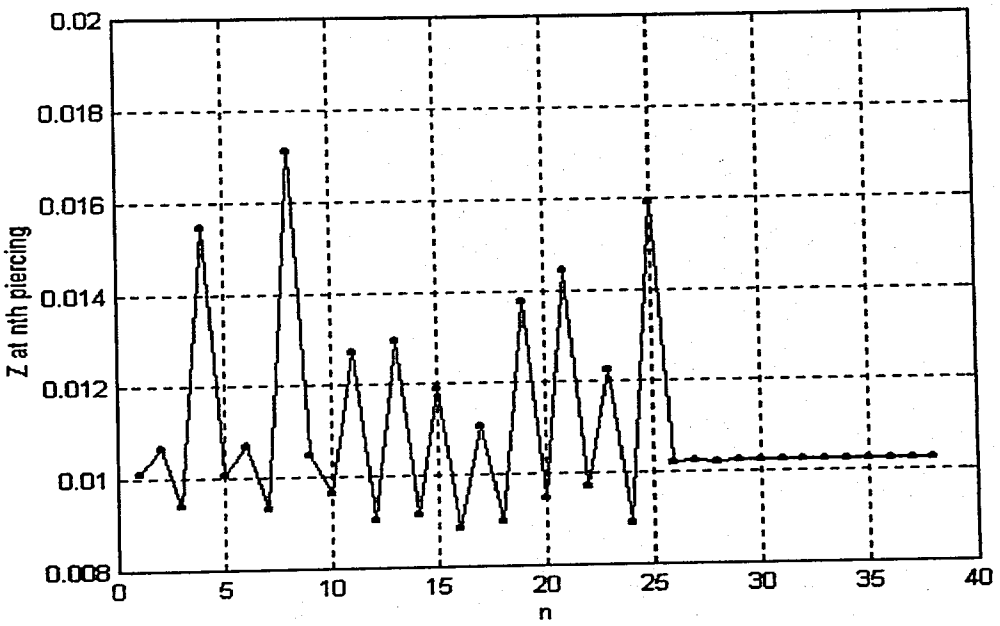




FIGURE 4.9. States of the Rossler system controlled by the RS method

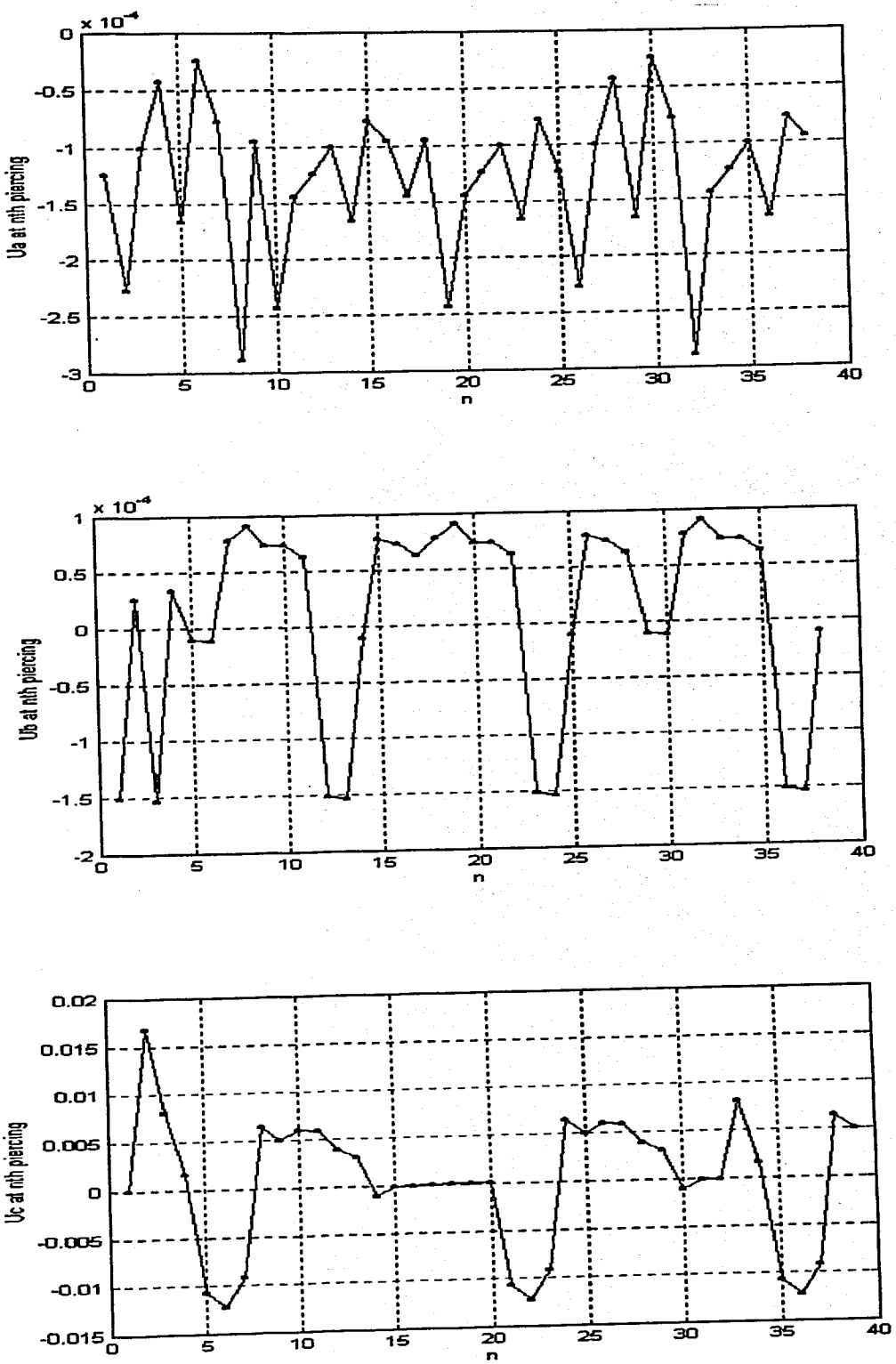



FIGURE 4.10. Change in control parameters about their nominal values for the Rossler system controlled by the RS method

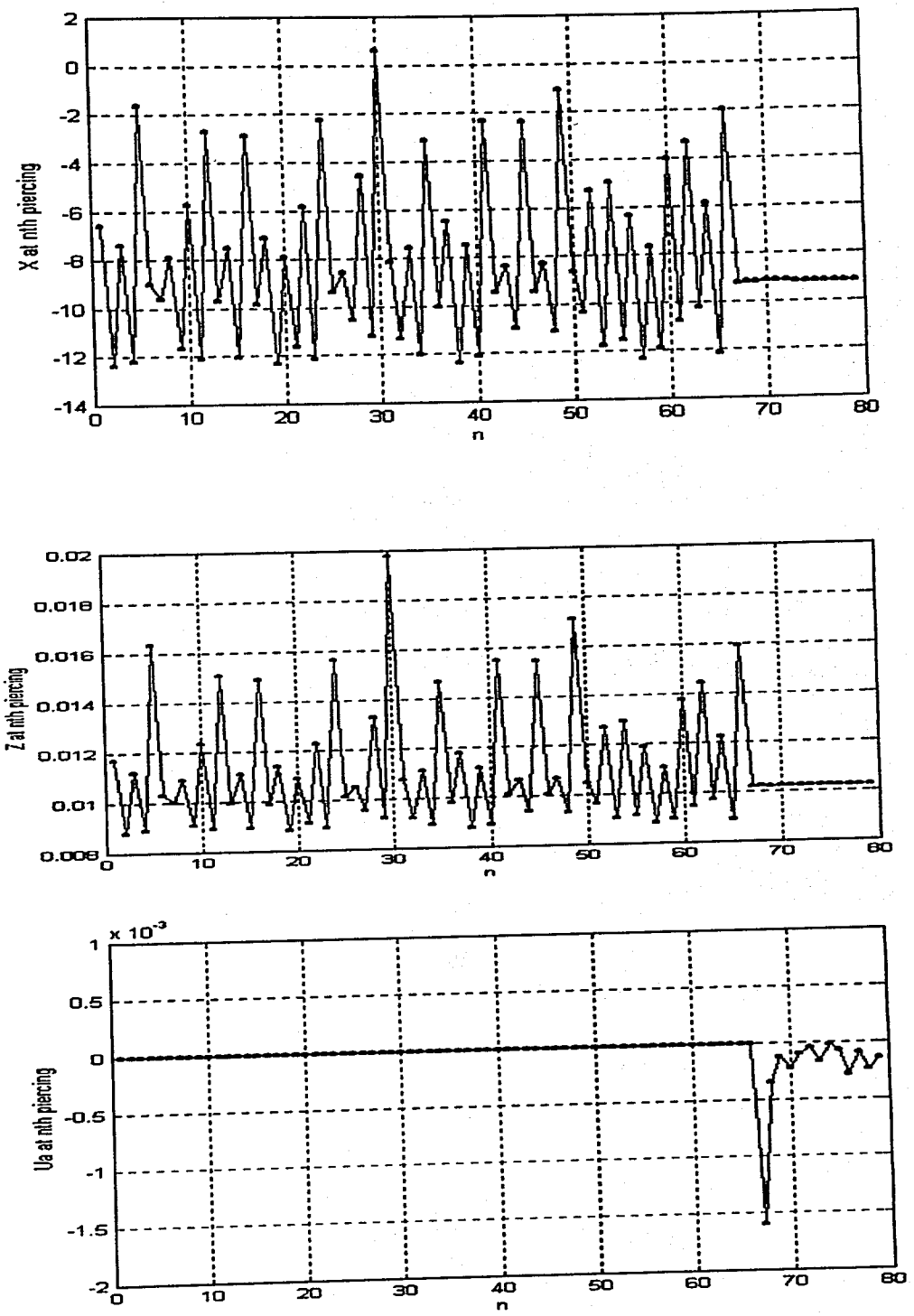



FIGURE 4.11. Change in control parameters about their nominal values and the states of the Rossler system controlled by the OGY method

TABLE 4.2. Comparison of the controllers for the Rossler system

| | The OGY Method | The NN Based ECR | The RS Method |
|---|---|---|---|
| Number of control regions | 0 | 1 | 10 |
| Radius of the target region ($\delta$) | 0.3 | 0.3 | 0.3 |
| Max. Allowed par. change ($\delta a_{max}$) | 0.004 | 0.004 | 0.004 |
| Max. Allowed par. change ($\delta b_{max}$) | 0.0006 | 0.0006 | 0.0006 |
| Max. Allowed par. change ($\delta c_{max}$) | 0.03 | 0.03 | 0.03 |
| Training set (patterns) | 142 | 275 | 238 |
| Training time (CPU time) | 72560 | 98970000 | 967380000 |
| Average deviation from target | 0.072 | 0.091 | 0.05246 |
| Average deviation from $a_{nom}$ | 0.000248 | 0.000154 | 0.000248 |
| Average deviation from $b_{nom}$ | 0.0 | 0.0000301 | 0.0000814 |
| Average deviation from $c_{nom}$ | 0.0 | 0.0044 | 0.00643 |
| Avr. Reaching time(piercing) | 66.81 | 29.31 | 26.17 |

### 4.4.3. The Hénon Map

Stabilisation of the equilibrium point of the Hénon map is achieved by the OGY method, the Neural Network Based ECR method and the RS method. In Figure 4.12, control system for the Hénon map is illustrated, where ($x_d$, $y_d$) is the equilibrium point vector of the Hénon map or the desired vector on any region. For these three controllers, the numerical results are given in the Table 4.3. The outputs of the Hénon map controlled by the controllers can be seen from Figure 4.13 to Figure 4.14.

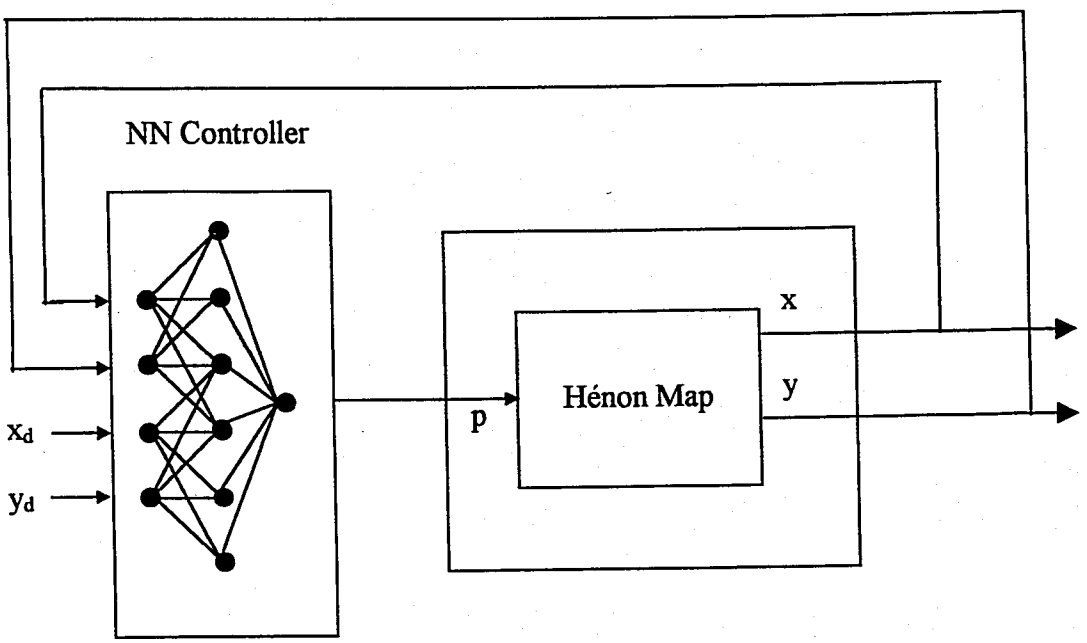



FIGURE 4.12. Control system for the Hénon map

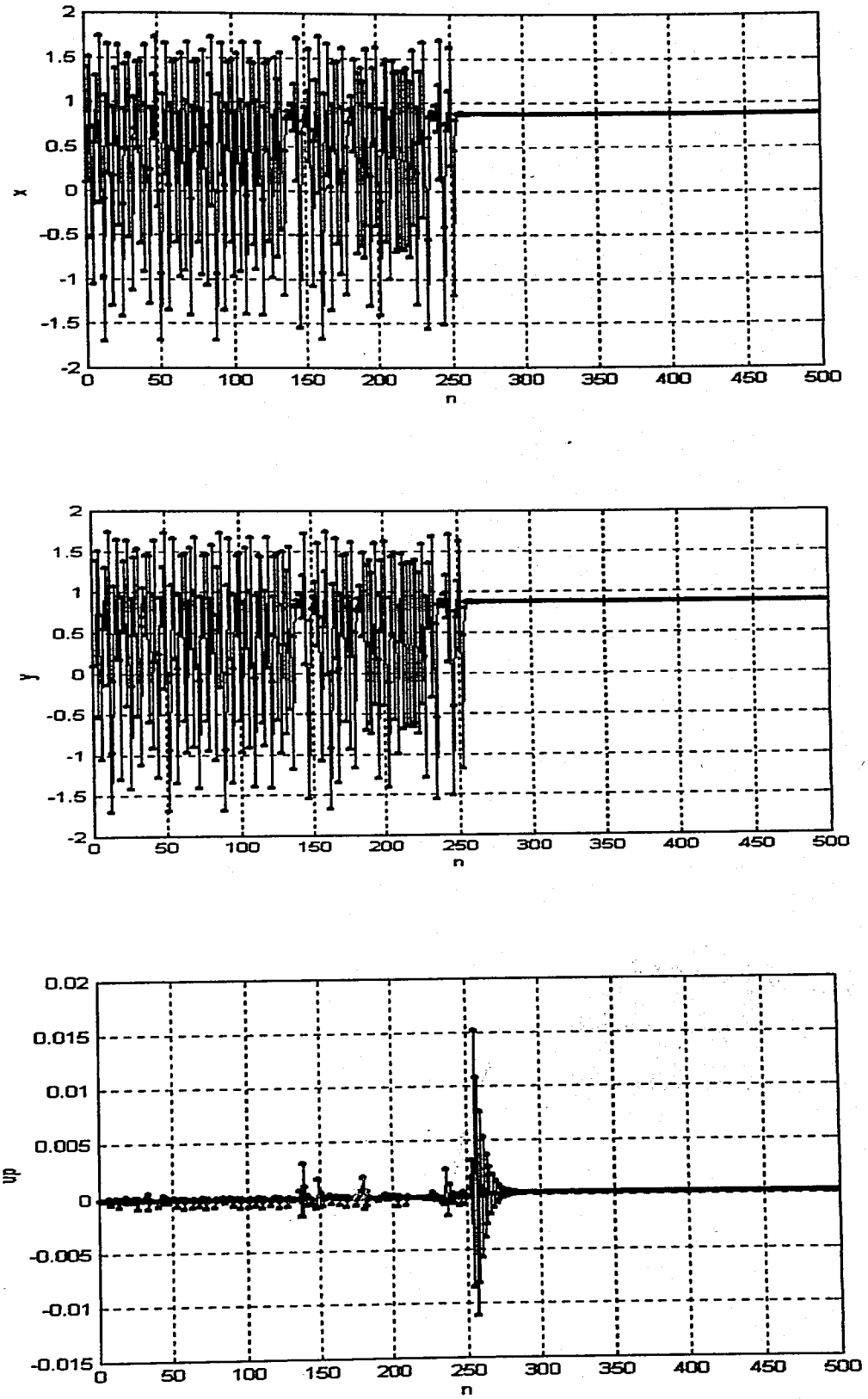



FIGURE 4.13. Change in control parameter about its nominal value and the states of the Hénon map controlled by the RS method.

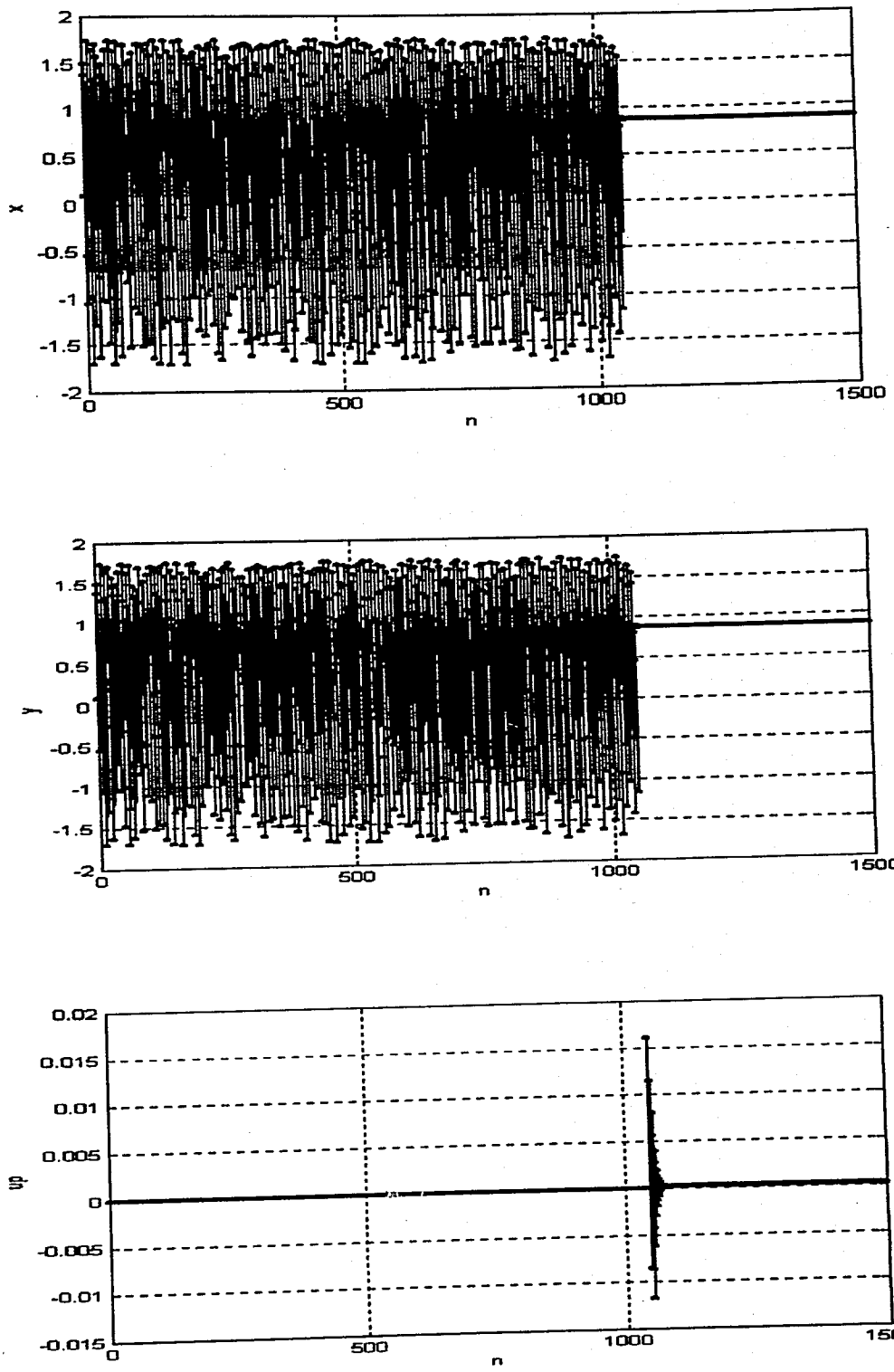



FIGURE 4.14. Change in control parameter about its nominal value and the states of the Hénon map controlled by the OGY method.

TABLE 4.3. Comparison of the controllers for the Hénon map

| | The OGY Method | The NN Based ECR | The RS Method |
|---|---|---|---|
| Number of activation regions | 0 | 3 | 3 |
| Radius of the control region ($\delta$) | 0.02 | 0.02 | 0.02 |
| Max. Allowed par. Change ($\delta p_{max}$) | 0.03 | 0.03 | 0.03 |
| Training set (patterns) | 116 | 172 | 300 |
| Training time (CPU time) | 5814 | 34640000 | 103086000 |
| Average deviation from target | 0.0080 | 0.0010 | 0.0002 |
| Average deviation from $p_{nom}$ | 0.00010570 | 0.00154800 | 0.00044719 |
| Avr. reaching time (iterations) | 767.9 | 103.9 | 119.97 |

### 4.4.4. The Logistic Map

Stabilisation of the equilibrium point of the Logistic map is achieved by the OGY method, the Neural Network Based ECR method and the RS method. In Figure 4.15, control system for the Logistic map is illustrated, where $x_d$ is the equilibrium point of the Logistic map or the desired point on any region. For these three controllers, the numerical results are given in the Table 4.4. The outputs of the Logistic map controlled by the controllers can be seen from Figure 4.16 and Figure 4.17.

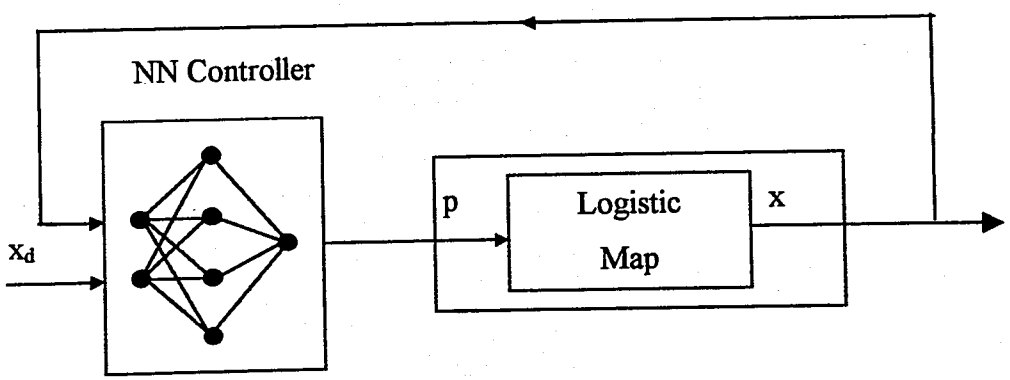



FIGURE 4.15. Control system for the Logistic map

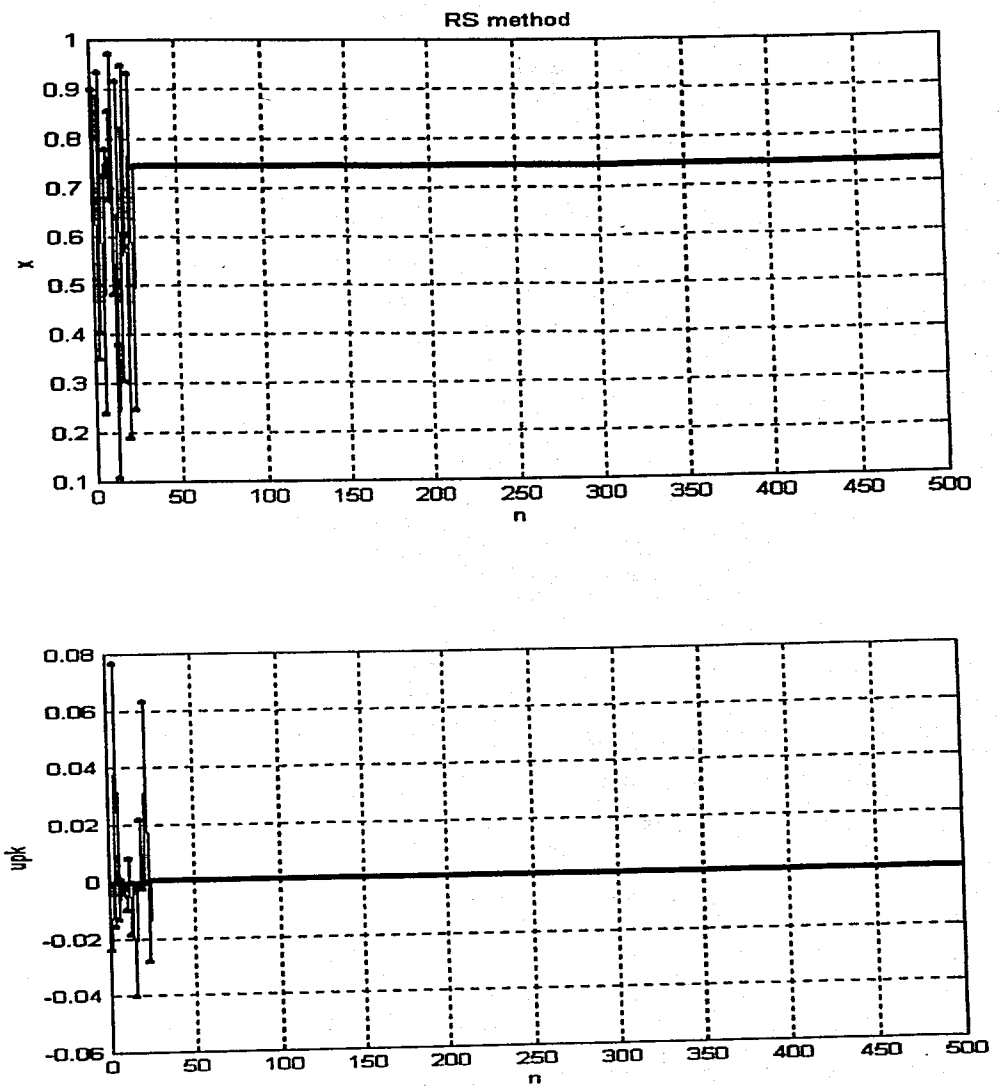



FIGURE 4.16. Change in control parameter about its nominal value and the state of the Logistic map controlled by the RS method.

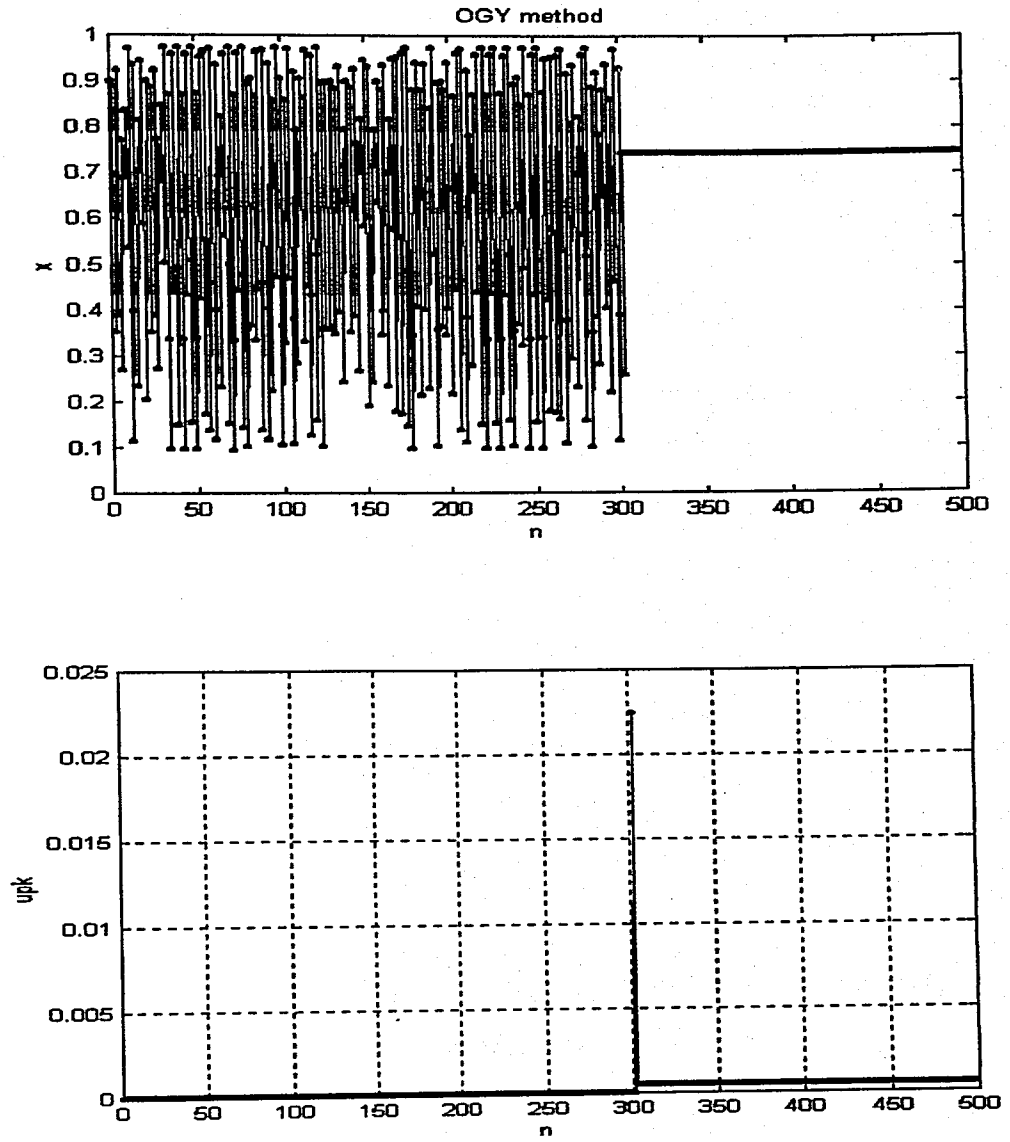



FIGURE 4.17. Change in control parameter about its nominal value and the state of the Logistic map controlled by the OGY method.

TABLE 4.4. Comparison of the controllers for the Logistic map

| | The OGY Method | The NN Based ECR | The RS Method |
|---|---|---|---|
| Number of activation regions | 0 | 2 | 2 |
| Radius of the control region ($\delta$) | 0.01 | 0.01 | 0.01 |
| Max. allowed par. change ($\delta p_{max}$) | 0.1 | 0.1 | 0.1 |
| Training set (patterns) | 83 | 141 | 200 |
| Training time (CPU time) | 2199 | 18900000 | 21700000 |
| Average deviation from target | 0.00003063 | 0.002015 | 0.00003484 |
| Average deviation from $p_{nom}$ | 0.0003075 | 0.03089 | 0.0003313 |
| Avr. reaching time (iterations) | 143.7 | 48.25 | 23.25 |

# 5. EVALUTION OF THE RESULTS AND CONCLUSIONS

The following aspects are considered evaluating and comparing different chaos control approaches based on the OGY method.

(a) *Dynamics of the System:* Since the governing equations of a chaotic system are rather difficult to obtain, whether the controller requires any a priori knowledge about the system dynamics or not, is an important comparison metric in the chaos control literature. In the method proposed in this thesis, the knowledge about the system dynamics is not required. The set of observed data is used to model the strange attractor as a FSM.

(b) *Targeting*: To drive the trajectories to the target region, targeting task has to be fulfilled. To reduce the average reaching time, obviously the RS method provides an appreciable improvement. It reduces the average reaching time by 50% up to 75% depending on the system.

(c) *The Amount of Data Needed:* The number of data required to identify system dynamics under consideration is preferred to be small. But the RS method requires data on middle level compared to other methods.

(d) *Training Time:* The time required to model or to identify the system dynamics should be as short as possible. As can be seen in Section 4, the conventional OGY method exclusively gives the best performance since the training is based on the local linear approximation of the dynamics in the sense of least squares, whereas the training of the neural networks obviously takes longer time.

(e) *Control Effort:* The control effort needed for satisfactory control performance is an important criterion that should be evaluated. The average deviation from the nominal parameter value, which is a measure for the control effort, the neural network based controllers give the best performance for most of the systems. Furthermore, amplitude of the control parameter is guaranteed to be small.

(f) *Multi-Parameter Control:* Most of the chaotic systems have more than one control parameter, and in some cases adjusting only one parameter may not be able to achieve the desired control task or the amplitude of the control parameter can not be limited in a small range. Thus, controllers applicable to the multi-parameter control are preferable. From this point of view, RS method can be used in most of the systems

(g) *Higher Dimensional Systems*: Generally, most of the chaotic dynamics have dimensions higher than one, i.e. for flows, the dimension of the system has to be greater than two to observe chaotic phenomena, thus the applicability to higher dimensional systems is an important feature. The RS method also accomplishes this property.

(h) *Quality of Control* : Small deviations from the target after the system converges determines the quality of the control. The neural network based methods perform better than the OGY method due to the high approximation capability and fault tolerant architecture of neural networks.

The evaluation of conventional OGY, the Neural Network Based ECR and the RS methods with respect to some criteria are given in the Table 5.1 where the controller that gives the best performance with respect to each criterion considered is marked.

TABLE 5.1. Overall comparison of the controllers

| | | OGY Method | ECR Method | RS Method |
|---|---|---|---|---|
| The Logistic Map | Training time | X | | |
| | Average deviation from the target | X | | |
| | Avr. Dev. from the nom. parameters | X | | |
| | Average reaching time | | | X |
| The Henon Map | Training time | X | | |
| | Average deviation from the target | | | X |
| | Avr. Dev. from the nom. parameters | X | | |
| | Average reaching time | | X | |
| The Lorenz System | Training time | X | | |
| | Average deviation from the target | | | X |
| | Avr. Dev. from the nom. parameters | | | X |
| | Average reaching time | | | X |
| The Rossler System | Training time | X | | |
| | Average deviation from the target | | | X |
| | Avr. Dev. from the nom. parameters | | | X |
| | Average reaching time | | | X |

As a result of this study, one can conclude that the choice of the appropriate controller depends on the application area and performance specifications. As the results indicate that, for complicated cases, e.g. higher dimensions, the RS method is preferred with respect to various criteria. Furthermore, when fast reaching time is required RS method can be strongly recommended. It is obvious that, if the short training time is important and the targeting issue is not considered the classical OGY method gives the better results.

When compared to other chaos control methods existing in the literature, the RS method still preserves its superiority with respect to small reaching time and meets other performance criteria, as can be observed in Table 5.2, with the exception of on-line training facility.

TABLE 5.2. An overview of some controllers used for chaotic dynamics

(The methods considered in this table does not require the a priori knowledge of system dynamics and realise the control task by adjusting the control parameter(s).)

| | Training time | Amount of data required | Multi-parameter control | Higher dimension | Targeting or local control | On-line or off-line control | Control effort | Response |
|---|---|---|---|---|---|---|---|---|
| The original OGY | Very short | Little | Only one | Up to 2-D | Local | Off-line | Low | Very fast |
| The Extended OGY in [41,42] | Short | Middle | Available | Available. | Local | Off-line | Low | Very fast |
| The Extended OGY in [40] | Middle | Middle | Available | Available | Both | On-line | High | Very fast |
| The Self Continuous Control | Very Short | Little | Only one | Up to 3-D | Both | Off-line | High | Very fast |
| The Neural Network Based ECR | Middle | Middle | Available | Available | Both | Off-line | Low | Fast |
| Region Scheduling (RS) Method | Very long | Middle | Available | Available | Both | Off-line | Low | Fast |

# 6. SUGGESTIONS FOR FURTHER WORK

The results obtained so far indicate that the proposed RS method gives a promising performance. Further work on this method should include simulations and experimental studies on higher dimensional systems to verify the results.

Since real applications cannot be free from noise, the RS controller should have sufficient robustness against noise.

Future works on the RS method should also concentrate on the intelligent partitioning of the strange attractor into regions since it was applied in a heuristic manner. Some methods have to be developed in order to determine the number and the locations of the regions from the empirical data.

# APPENDIX A

MatLab Codes Simulate the Chaotic Systems

```
%~~~~~~~~~~~~~~~~~
%~~~Lorenz System
%~~~~~~~~~~~~~~~~~
DELTA = 0.001;

s = 10.0;
r = 28.0;
b = 8.0/3.0;

xk = X0(1);
yk = X0(2);
zk = X0(3);
us = X0(4);
itr = X0(5);

k = 1;
t = 1;

while(k<itr+1)
  k1x = DELTA * ((s+us)*(yk-xk));
  k1y = DELTA * (r*xk-yk-xk*zk);
  k1z = DELTA * (xk*yk-b*zk);

  k2x = DELTA * ((s+us)*((yk+k1y/2.0)-(xk+k1x/2.0)));
  k2y = DELTA * (r*(xk+k1x/2.0)-(yk+k1y/2.0)-(xk+k1x/2.0)*(zk+k1z/2.0));
  k2z = DELTA * ((xk+k1x/2.0)*(yk+k1y/2.0)-b*(zk+k1z/2.0));

  k3x = DELTA * ((s+us)*((yk+k2y/2.0)-(xk+k2x/2.0)));
```

```
k3y = DELTA * (r*(xk+k2x/2.0)-(yk+k2y/2.0)-(xk+k2x/2.0)*(zk+k2z/2.0));
k3z = DELTA * ((xk+k2x/2.0)*(yk+k2y/2.0)-b*(zk+k2z/2.0));

k4x = DELTA * ((s+us)*((yk+k3y/2.0)-(xk+k3x/2.0)));
k4y = DELTA * (r*(xk+k3x/2.0)-(yk+k3y/2.0)-(xk+k3x/2.0)*(zk+k3z/2.0));
k4z = DELTA * ((xk+k3x/2.0)*(yk+k3y/2.0)-b*(zk+k3z/2.0));

xkp1 = xk+k1x/6.0+k2x/3.0+k3x/3.0+k4x/6.0;
ykp1 = yk+k1y/6.0+k2y/3.0+k3y/3.0+k4y/6.0;
zkp1 = zk+k1z/6.0+k2z/3.0+k3z/3.0+k4z/6.0;

if((yk>8.47) & (ykp1<8.47))
  gain  = (8.47-ykp1) / (yk-ykp1);

  xp(k) = xkp1+gain*(xk-xkp1);
  zp(k) = zkp1+gain*(zk-zkp1);
  timeP(k) = t;
  k = k+1;
 end
 xk = xkp1;
 yk = ykp1;
 zk = zkp1;
 t = t+1;
end

%~~~~~~~~~~~~~~~~
%~~~Rossler System
%~~~~~~~~~~~~~~~~

DELTA = 0.001;

a = 0.150;
b = 0.20;
```

```
c = 10.0;

xk = X0(1);
yk = X0(2);
zk = X0(3);

ua = X0(4);
ub = X0(5);
uc = X0(6);

itr = X0(7);

k = 1;
t = 1;

while(k<itr+1)
  k1x = DELTA * (-yk-zk);
  k1y = DELTA * (xk+(a+ua)*yk);
  k1z = DELTA * ((b+ub) + (xk-(c+uc)) * zk);

  k2x = DELTA * (-(yk+k1y/2)-(zk+k1z/2));
  k2y = DELTA * ((xk+k1x/2)+(a+ua)*(yk+k1y/2));
  k2z = DELTA * ((b+ub) + ((xk+k1x/2)-(c+uc)) * (zk+k1z/2));

  k3x = DELTA * (-(yk+k2y/2)-(zk+k2z/2));
  k3y = DELTA * ((xk+k2x/2)+(a+ua)*(yk+k2y/2));
  k3z = DELTA * ((b+ub) + ((xk+k2x/2)-(c+uc)) * (zk+k2z/2));

  k4x = DELTA * (-(yk+k3y/2)-(zk+k3z/2));
  k4y = DELTA * ((xk+k3x/2)+(a+ua)*(yk+k3y/2));
  k4z = DELTA * ((b+ub) + ((xk+k3x/2)-(c+uc)) * (zk+k3z/2));

  xkp1 = xk+k1x/6.0+k2x/3.0+k3x/3.0+k4x/6.0;
```

```
ykp1 = yk+k1y/6.0+k2y/3.0+k3y/3.0+k4y/6.0;
zkp1 = zk+k1z/6.0+k2z/3.0+k3z/3.0+k4z/6.0;

if((yk>-8.0) & (ykp1<-8.0))
  xp(k) = xk;
  yp(k) = yk;
  zp(k) = zk;

  timeP(k) = t;
  k = k+1;
 end
 xk = xkp1;
 yk = ykp1;
 zk = zkp1;
 t = t+1;
end

%~~~~~~~~~~~~~~~~
%~~~Hénon Map
%~~~~~~~~~~~~~~~~

clear all

p = 1.37;

xk = 0.001;
yk = 0.001;
up = 0;
itr = 300;

n = 1;
while(n<itr)
  xkp1 = (p+up)+0.3*yk-xk^2;
```

```
ykp1 = xk;

% save states
x(n) = xk;
y(n) = yk;
xp1(n) = xkp1;
yp1(n) = ykp1;

xk = xkp1;
yk = ykp1;
n = n+1;
end

save henmap.mat xp1 yp1 x y;


%~~~~~~~~~~~~~~~
%~~~Logistic Map
%~~~~~~~~~~~~~~~

X0 = [0.001 0 100];

p = 3.9;

xk = X0(1);
up = X0(2);
itr = X0(3);

n = 1;
while(n<itr)
   xkp1 = (p+up)*xk*(1-xk);

   x(n) = xk;
```

```
    xp1(n) = xkp1;

    xk = xkp1;
    n = n+1;
end

save logmap.mat xp1 x;
```

# APPENDIX B

MatLab Codes Implement The OGY Control Algorithm

*These codes are the implementation of the OGY method for the Lorenz system*

```
clear all
```

*loading the file contains the data obtained for the nominal parameter values*

```
load ldat01dc.mat
```

*find the equilibrium point*

```
[a b] = min(abs(dc1(1:length(dc1)-1)-dc1(2:length(dc1)))+abs(dc2(1:length(dc2)-
1)-dc2(2:length(dc2)))+abs(dc3(1:length(dc3)-1)-dc3(2:length(dc3))));
dc1fx=dc1(b);
dc2fx=dc2(b);
dc3fx=dc3(b);
clear dc1;clear dc2;clear dc3;
```

*loading the file contains the data obtained for the parameter values changed within their allowable range*

```
load ldat02dc.mat
delta=0.3; δ, radius of the local control region
DC1rms=sqrt(sum(dc1.^2)/length(dc1));RMS value of the delay coordinate 1
noiserate=0*DC1rms;
```

*noise is being added*

```
dc1=dc1+2*(rand-0.5)*noiserate;
dc2=dc2+2*(rand-0.5)*noiserate;
dc3=dc3+2*(rand-0.5)*noiserate;
```

*Collecting the input output pairs in the $\delta$ neighbourhood of the equilibrium point*

```
[I Y ]=find((abs(dc1-dc1fx)<delta)&(abs(dc2-dc2fx)<delta)&(abs(dc3-
dc3fx)<delta));
D(I)=ones(1,length(I));
[L M]=find( D(1:length(D)-1)==1 & D(2:length(D))==1);clear L;
I=M;
NOP=length(I);
```

*The map is obtained as below*

```
%.....%
%[DZn+1]=[DZn][A]+[Dpn-1][B]+[Dpn][C]
%.....%
        U1=dc1(I+1)-dc1fx;
        D1(1:NOP,1)=dc1(I)-dc1fx;
        D1(1:NOP,2)=dc2(I)-dc2fx;
        D1(1:NOP,3)=dc3(I)-dc3fx;
        D1(1:NOP,4)=ds(I-1);
        D1(1:NOP,5)=dr(I-1);
        D1(1:NOP,6)=db(I-1);
        D1(1:NOP,7)=ds(I);
        D1(1:NOP,8)=dr(I);
        D1(1:NOP,9)=db(I);

        U2=dc2(I+1)-dc2fx;
        D2(1:NOP,1)=dc1(I)-dc1fx;
        D2(1:NOP,2)=dc2(I)-dc2fx;
        D2(1:NOP,3)=dc3(I)-dc3fx;
        D2(1:NOP,4)=ds(I-1);
        D2(1:NOP,5)=dr(I-1);
        D2(1:NOP,6)=db(I-1);
        D2(1:NOP,7)=ds(I);
        D2(1:NOP,8)=dr(I);
```

```
D2(1:NOP,9)=db(I);

U3=dc3(I+1)-dc3fx;
D3(1:NOP,1)=dc1(I)-dc1fx;
D3(1:NOP,2)=dc2(I)-dc2fx;
D3(1:NOP,3)=dc3(I)-dc3fx;
D3(1:NOP,4)=ds(I-1);
D3(1:NOP,5)=dr(I-1);
D3(1:NOP,6)=db(I-1);
D3(1:NOP,7)=ds(I);
D3(1:NOP,8)=dr(I);
D3(1:NOP,9)=db(I);

clear dc1;clear dc2;clear dc3;clear ds;clear dr;clear db;

V1=inv(D1'*D1)*D1'*U1;
V2=inv(D2'*D2)*D2'*U2;
V3=inv(D3'*D3)*D3'*U3;

V=([V1,V2,V3]);clear V1;clear V2;

A=V(1:3,1:3);
B=V(4:6,1:3);
C=V(7:9,1:3);
```

*Proper parameter changes are computed as follows;*

$$\mathbf{Dp}_n=-\mathbf{DZ}_n\mathbf{AC}^T(\mathbf{CC}^T)^{-1}-\mathbf{Dp}_{n-1}\mathbf{BC}^T(\mathbf{C}.\,\mathbf{C}^T)^{-1}$$

**P.S.** The codes for the other chaotic systems are the same as those for the Lorenz system except for the dimensions of the matrices **A**, **B**, and **C**. If the previous parameter change is not considered the matrix **C** is omitted

# APPENDIX C

MatLab Codes Implement The RS Method

*These codes are the implementation of the RS method for the Lorenz system*

```
clear all;

DELTA = 0.0010;
SURFACE = 8.48528137423857; % = sqrt(72)

s = 10.0;
r = 28.0;
b = 8.0/3.0;

xf = 14.23873330496232;
yf = 8.42508333328811;
zf = 39.79338962870591;

% chosen points from the regions 2.,3. ve 4 (look-up table)
xf2 = 10.50232881736006;
yf2 = 8.48226344993244;
zf2 = 32.25433421599135;

xf3 =  11.75541586455484;
yf3 = 8.42158648798486;
zf3 = 34.31299471404960;

xf4 = 12.18706858714795;
yf4 = 8.39003175019340;
zf4 = 34.99971846040494;

%xk = 0.001;
```

```
%yk = 0.001;
%zk = 0.001;

x0 = [7.96320550981350  -5.14080231036927  21.54515581352721];
xk = x0(1);
yk = x0(2);
zk = x0(3);% >0

xrmin = 9.0;
stepsize = 0.92846942739868;

us = 0.0;
ur = 0.0;
ub = 0.0;

k = 1;
t = 1;

 while(k<25)
   k1x = DELTA * ((s+us)*(yk-xk));
   k1y = DELTA * ((r+ur)*xk-yk-xk*zk);
   k1z = DELTA * (xk*yk-(b+ub)*zk);

   k2x = DELTA * ((s+us)*((yk+k1y/2.0)-(xk+k1x/2.0)));
   k2y = DELTA * ((r+ur)*(xk+k1x/2.0)-(yk+k1y/2.0)-(xk+k1x/2.0)*(zk+k1z/2.0));
   k2z = DELTA * ((xk+k1x/2.0)*(yk+k1y/2.0)-(b+ub)*(zk+k1z/2.0));

   k3x = DELTA * ((s+us)*((yk+k2y/2.0)-(xk+k2x/2.0)));
   k3y = DELTA * ((r+ur)*(xk+k2x/2.0)-(yk+k2y/2.0)-(xk+k2x/2.0)*(zk+k2z/2.0));
   k3z = DELTA * ((xk+k2x/2.0)*(yk+k2y/2.0)-(b+ub)*(zk+k2z/2.0));

   k4x = DELTA * ((s+us)*((yk+k3y/2.0)-(xk+k3x/2.0)));
   k4y = DELTA * ((r+ur)*(xk+k3x/2.0)-(yk+k3y/2.0)-(xk+k3x/2.0)*(zk+k3z/2.0));
```

```
k4z = DELTA * ((xk+k3x/2.0)*(yk+k3y/2.0)-(b+ub)*(zk+k3z/2.0));

xkp1 = xk + k1x/6.0 + k2x/3.0 + k3x/3.0 + k4x/6.0;
ykp1 = yk + k1y/6.0 + k2y/3.0 + k3y/3.0 + k4y/6.0;
zkp1 = zk + k1z/6.0 + k2z/3.0 + k3z/3.0 + k4z/6.0;

if((yk>SURFACE) & (ykp1<SURFACE))
  xp(k) = xkp1;
  yp(k) = ykp1;
  zp(k) = zkp1;
  timeP(k) = t;
  inreg(k) = 0;
  inogy(k) = 0;

  dist = sqrt((xp(k)-xf)^2+(yp(k)-yf)^2+(zp(k)-zf)^2);
  if(dist< 0.3)
  %dist = abs(xp(k)-xf);
  %if(dist< 0.2)
    neuInput = [xp(k) yp(k) zp(k) xf yf zf]';

    inreg(k) = 6;
    inogy(k) = 1;

    load rbfpar.mat
    u  = (rbfdesig(neuInput,c(:,subset),rr,'m')*w);
    clear  H c rr w subset

    us = u(1) - 10.0;
    ur = u(2) - 28.0;
    ub = u(3) - 8.0/3.0;

    if(abs(us)>0.3)
      us = 0;
```

```
      ur = 0;
      ub = 0;
     disp('saturation us in OGY');
   end
   if(abs(ur)>0.84)
     us = 0;
      ur = 0;
      ub = 0;
     disp('saturation ur in OGY');
   end
   if(abs(ub)>0.08)
     us = 0;
      ur = 0;
      ub = 0;
     disp('saturation ub in OGY');
   end

  else
%---find region
        j = 1;
                regnum = 0;
         while(regnum == 0)
         if(xp(k) <= (xrmin+j*stepsize))
        regnum = j;
         end
         j = j + 1;
         end
     if(j>10)
      regnum = 0;
     end
        if(regnum ~= 0)
      inreg(k) = regnum;
      eval(['load rbf' num2str(regnum) 'par.mat']);
```

```
  if((regnum == 1) | (regnum == 2) | (regnum == 3))
        eval(['neuInput = [xp(k) yp(k) zp(k)  xf' num2str(regnum+1) ' yf'
        num2str(regnum+1) ' zf' num2str(regnum+1) ']';']);
  neuInput = neuInput';
else
    neuInput = [xp(k) yp(k) zp(k) xf yf zf]';
end
u  = (rbfdesig(neuInput,c(:,subset),rr,'i')*w);
clear  H c rr w subset

us = u(1) - 10.0;
ur = u(2) - 28.0;
ub = u(3) - 8.0/3.0;

if(abs(us)>0.3)
   us = 0;
   ur = 0;
   ub = 0;
   disp('saturation us');disp(regnum);
 end
 if(abs(ur)>0.84)
                                                  us = 0;
    ur = 0;
    ub = 0;
    disp('saturation ur');disp(regnum);
 end
 if(abs(ub)>0.08)
    us = 0;
    ur = 0;
    ub = 0;
    disp('saturation ub');disp(regnum);
 end
```

```
    else
        us = 0.0;
      ur = 0.0;
      ub = 0.0;
    end
  end
  usk(k) = us;
  urk(k) = ur;
  ubk(k) = ub;

  k = k+1;
 end
 xk = xkp1;
 yk = ykp1;
 zk = zkp1;
 t = t+1;
end
```

**P.S.** The codes for the other chaotic systems are similar as those for the Lorenz system.

# REFERENCES


1. Mandelbrot B. B., "The fractal geometry of Nature", W.H. Freeman, 1982

2. Ott, E., C. Grebogi, and J. Yorke, "Controlling Chaos," *Physical Review Letters,* Vol. 64, No. 11, pp. 1196-1199, March 1990.

3. Ditto, W. L., S. N. Rauseo, and M. L. Spano, "Experimental Control of Chaos," *Physical Review Letters,* Vol. 65, No. 26, pp. 3211-3214, December 1990.

4. Hunt, E. R., "Stabilizing High-period Orbits in a Chaotic System: The Diode Resonator," *Physical Review Letters,* Vol. 67, No. 15, pp. 1953-1955, October 1991.

5. Shinbrot, T., C. Grebogi, E. Ott, and J. A. Yorke, "Using Small Perturbations to Control Chaos," *Nature,* Vol. 363, pp. 411-417, June 1993.

6. Bielawski, S., D. Derozier, and P. Glorieux, "Experimental Chracterization of Unstable Periodic Orbits By Controlling Chaos," *Physical Review E,* vol. 47, No. 4, pp. 2492-2495, April 1993.

7. Gills, Z., C. Iwata, R. Roy, I. B. Schwartz, and I. Triandaf, "Tracking Unstable Steady States: Extending the Stability Regime of a Multimode Laser System," *Physical Review Letters,* Vol. 69, No. 22, pp. 3169-3172, November 1992.

8. Auerbach, D., C. Grebogi, E. Ott, and J. A. Yorke, "Controlling Chaos in High Dimensional Systems," *Physical Review Letters,* Vol. 69, No. 24, pp. 3479-3482, December 1992.

9. Shinbrot, T., E. Ott, C. Grebogi, and J. A. Yorke, "Using Chaos to Direct Trajectories to Targets," *Physical Review Letters,* Vol. 65, No. 26, pp. 3215-3218, December 1990.

10. Shinbrot, T., W. Ditto, C. Grebogi, E. Ott, M. Spano, and J. A. Yorke, "Using the Sensitive Dependence of Chaos (the "Butterfly Effect") to Direct Trajectories in an Experimental Chaotic System," *Physical Review Letters,* Vol. 68, No. 19, pp. 2863-2866, May 1992.

11. Romeiras, F., C. Grebogi, E. Ott, and W. P. Dayawansa, "Controlling Chaotic Dynamical Systems," *Physica D,* Vol. 58, pp. 165-192, 1992.

12. Auerbach, D., C. Grebogi, E. Ott, and J. A. Yorke, "Controlling Chaos in High Dimensional Systems," *Physical Review Letters,* Vol. 69, No. 24, pp. 3479-3482, December 1992.

13. Nitsche, G., and U. Dressler, "Controlling Chaotic Dynamical Systems Using Time Delay Coordinates," *Physica D,* Vol. 58, pp. 153-164, 1992.

14. Rollins, R. W., P. Parmananda, and P. Sherard, "Controlling Chaos in Highly Dissipative Systems: A Simple Recursive Algorithm," *Physical Review E,* Vol. 47, No. 2, pp. 780-787, February 1993.

15. Parmananda, P., P. Sherard, R. W.Rollins, and H. D. Dewald, "Control of Chaos in an Electrochemical Cell," *Physical Review E,* Vol. 47, No. 5, pp. 3003-3006, May 1993.

16. Lai, Y., M. Ding, and C. Grebogi, "Controlling Hamiltonian Chaos," *Physical Review E,* Vol. 47, No. 1, pp. 86-92, January 1993.

17. Barreto, E., and C. Grebogi, "Multiparameter Control of Chaos," *Physical Review E,* Vol. 52, No. 4, pp. 3553-3557, October 1995.

18. So, P., and E. Ott, "Controlling Chaos Using Time Delay Coordinates via Stabilization of Periodic Orbits," *Physical Review E,* Vol. 51, No. 4, pp. 2955-2962, April 1995.

19. Ding, M., W. Yang, V. In, W. L. Ditto, M. L. Spano, and B. Gluckman, "Controlling Chaos in High Dimensions: Theory and Experiment," *Physical Review E,* Vol. 53, No. 5, pp. 4334-4344, May 1996.

20. Rhode, M. A., J. Thomas, R. W. Rollins, and A. J. Markworth, "Automated Adaptive Recursive Control of Unstable Orbits in High-Dimensional Chaotic Systems," *Physical Review E,* Vol. 54, No. 5, pp. 4880-4887, November 1996.

21. Gluckman, B. J., M. L. Spano, W. Yang, M. Ding, V. In, and W. L. Ditto, "Tracking Unstable Periodic Orbits in Nonstationary High-Dimensional Chaotic Systems: Method and Experiment," *Physical Review E* Vol.55, No. 5, pp. 4935-4942 May 1997.

22. Peng, B., V. Petrov, and K. Showalter, "Controlling Chemical Chaos," *Journal of Physical Chemistry,* Vol. 95, No. 13, pp. 4957-4959, 1991.

23. Pyragas, K., "Continuous Control of Chaos by Self-Controlling Feedback," *Physics Letters A,* Vol. 170, pp. 421-428, 1992.

24. Pyragas, K., and A. Tamasevicius, "Experimental Control of Chaos by Delayed Self-Controlling Feedback," *Physics Letters A,* Vol. 180, pp. 99-102, 1993.

25. Kittel, A., J. Parisi, and K. Pyragas, "Delayed feedback Control of Chaos by Self-Adapted Delay Time," *Physics Letters A,* Vol. 198, pp. 433-436, March 1995.

26. Qu, Z., G. Hu, and B. Ma, "Controlling Chaos via Continuous Feedback," *Physics Letters A,* Vol. 178, No. 3-4, pp. 265-270, July 1993.

27. Chen, Y. H., and M. Y. Chou, "Continuous feedback Approach for Controlling Chaos," *Physical Review E,* Vol. 50, No. 3, pp. 2331-2334, September 1994.

28. Barret, M. D., "Continuous Control of Chaos," *Physica D,* Vol. 91, pp. 340-348, 1996.

29. Wiesel, W. E., "Full Stability-Exponent Placement in Chaotic Systems," *Physical Review E,* Vol. 53, No. 2, pp. 1453-1458, February 1996.

30. Yu, X., "Controlling Lorenz Chaos, "*International Journal of Systems Science,* Vol. 27, pp. 355-359, 1996.

31. Konishi, K., and H. Kokame, "Stabilizing and Tracking unstable Focus Point in Chaotic Systems Using a Neural Network," *Physics Letters A*, Vol. 206, pp 203-210, 1995.

32. Konishi, K., and H. Kokame, "Control of Chaotic Systems Using an On-Line Trained Linear Neural Controller," *Physica D,* Vol. 100, pp 423-438, 1997.

33. Bakker, R., J. C. Schouten, F. Takens, and C. M. Bleek, "Neural Network Model to Control Experimental Chaotic Pendulum," *Physical Review E,* Vol. 54, No. 4, pp 3545-3552, October 1996.

34. Bakker, R., R. J. Korte, J. C. Schouten, F. Takens, and C. M. Bleek, "Neural Networks for Prediction and Control of Chaotic Fluized Bed Hydrodynamics: A First Step," *Fractals,* Vol. 5, No. 3, pp. 523-530, 1997.

35. Ogata K, "Control Engineering", second edition, Prentice-Hall, 1990

36. Gluckman, B. J., M. L. Spano, W. Yang, M. Ding, V. In, and W. L. Ditto, "Tracking Unstable Periodic Orbits in Nonstationary High-Dimensional Chaotic Systems: Method and Experiment," *Physical Review E* Vol.55, No. 5, pp. 4935-4942 May 1997.

37. S.Iplikci and Y. Denizhan, "Controlling Chaotic Dynamics via Neural Networks Using Radial Basis Functions", AIP Conference Proceedings. (Accepted for Publication)

38. Grimaldi, R. P., "Discrete and Combinatorial Mathematics", Addison-Wesley Publishing Company, 1989.

39. Henry D.I. Abarbanel, "Analysis of observed chaotic data", Springer-Verlag New York Inc., 1996.

40. Rhode, M. A., J. Thomas, R. W. Rollins, and A. J. Markworth, "Automated Adaptive Recursive Control of Unstable Orbits in High-Dimensional Chaotic Systems," *Physical Review E*, Vol. 54, No. 5, pp. 4880-4887, November 1996.

41. So, P., and E. Ott, "Controlling Chaos Using Time Delay Coordinates via Stabilization of Periodic Orbits," *Physical Review E*, Vol. 51, No. 4, pp. 2955-2962, April 1995.

42. Ding, M., W. Yang, V. In, W. L. Ditto, M. L. Spano, and B. Gluckman, "Controlling Chaos in High Dimensions: Theory and Experiment," *Physical Review E*, Vol. 53, No. 5, pp. 4334-4344, May 1996.

# REFERENCES NOT CITED

Strogatz, S. H., *Nonlinear Dynamics and Chaos,* Addison-Wesley Publishing Company,1994.

Khalil, H. K., *Nonlinear Systems,* Prentice Hall Inc., New Jersey, 1996.

Slotine, J. E. and W. Li, *Applied Nonlinear Control,* Prentice Hall Inc., New Jersey, 1991.

Thompson, J. M. T. and H. B. Stewart, *Nonlinear Dynamics and Chaos,* John Wiley and Sons Publishing Company, 1986.