Application of Chaotic Dynamics to Neural Information Processing

(カオス的ダイナミクスのニューラル情報処理への応用)

Isao TOKUDA

Contents

Chapter 1

Introduction

1.1 Motivation: Chaos in neural systems

Chaos is one of the greatest discoveries in the science and engineering of the 20th century [46, 81, 130, 136, 153, 190]. Striking feature of chaos is due to the fact that irregular complex dynamics can be generated from simple deterministic nonlinear system. Although such irregular dynamics had been studied in terms of stochastic dynamical processes, chaos has completely changed our view on nature. In fact, re-examinations of irregular dynamics in real world systems have shown that chaotic dynamics can be observed in many real systems of our science and engineering. The examples include fluid flows [74, 75, 170], chemical reactions [97, 153, 166], optical systems [4, 99], flooded ship motion [146, 190], electronic circuits [135], and many others.

Chaos can also be observed in many biological systems [45, 205] including the brain systems. In the learning process of rabbit olfactory system, Freeman *et al.* observed chaotic activity in the olfactory potential [59, 60, 61, 62, 63, 64, 183] and Babloyanz *et al.* [23, 24] analyzed human electro-encephalogram (EEG) in various mental stages and found low-dimensional chaos in several sleep stages. In single neuron level, Aihara *et al.* [8, 9, 11, 134] observed chaotic response in giant squid axon stimulated by periodic electric stimuli.

From the physiological observations that suggest chaotic activity both in single neuron level and in network level, it is a natural direction of the neural network studies to investigate possible functional roles of chaos in the brain. Freeman *et al.* suggested a functional role of chaos as a novelty filter when a rabbit memorizes a new odor [183, 216]. They also considered chaos as a nonlinear pattern classifier when a rabbit classifies memories of odors [64]. Babloyantz *et al.* [23] emphasized chaos as a resonance capacity amplifier against external stimuli. Tsuda *et al.* [202, 204], Aihara and Adachi [3, 5], and Nara and Davis [150, 151] considered chaos as a dynamical memory searcher.

Significance of chaos in neural systems has been also highlighted in neural networks applied to engineering problems. In fact, there are several inter-

esting research works reporting the enhancement of information processing capability of neural networks in the presence of chaotic dynamics. Lapedes and Farber [123] and Sato *et al.* [174, 175] trained several types of recurrent neural networks to learn chaotic dynamics and reported the enhanced learning capability in the neural networks. Nozawa [154] applied chaotic neural network [12] to solve the traveling salesman problem and reported that remarkable optimization capability can be observed when neural network exhibits chaotic behaviour. Chen and Aihara [35] realized a simulated annealing in chaotic neural networks and observed significantly good network performance for various optimization problems.

Engineering application is one of the most important research subjects in the neural networks. We also believe that significance of chaos in the brain can be found in the neural networks with engineering applications. Hence, **the present thesis aims to investigate possible functions of chaotic dynamics in neural information processing models in the light of their engineering applications.** The engineering applications include system identification, modeling, and search. Our subject studies will be briefly described in the next Sections.

1.2 Neural networks for engineering applications

In neural network research, two remarkable paradigms have been recently developed to expand the range of the neural network applications. One is the **back-propagation** learning [167, 168] and the other is the Hopfield **energy** [92, 93].

The *back-propagation* learning presents a framework of learning algorithm to teach feedforward neural networks to achieve desirable computational tasks. Since multi-layer feedforward neural networks have universal approximation capability [43, 67, 96], back-propagation neural networks have been applied to many engineering problems such as input-output modeling, system identification, time sequence recognition, and time series prediction [19, 52, 89, 145].

Hopfield [92, 93], on the other hand, realized an analogy between the Hebbian-type associative memory neural network [15, 16, 17, 20, 85, 119, 149] and the Ising spin model of the statistical physics and introduced the notion of *energy* to the associative memory neural network. The energy function not only provided us with a better understanding of the associative memory from the viewpoint of statistical physics [18, 89] but it also extended the applicability of associative memory neural network to engineering problems such as the optimization problem. In practice, Hopfield and Tank [94] defined a Lyapunov function for a class of neural networks with symmetric connections and showed that the network dynamics of energy descent property can be utilized to solve the traveling salesman problem. Hopfield-Tank approach to optimization problems has been examined and modified by many researchers [13, 86, 210] and widely applied to practical engineering problems.

This thesis focuses on the back-propagation learning and the Hopfield-Tank optimization technique as the typical engineering applications of neural networks. For these two applications, we introduce the idea of chaotic dynamics for enhancing the neural information processing capability and study the function of chaos in the neural networks. In the following Section, subject studies of each Chapter of this thesis are described.

1.3 Chaos for neural network applications

1.3.1 Modeling of chaotic dynamics

In **Chapter 2**, we study the problem for modeling chaotic dynamics by neural networks. Due to the universal approximation capability [43, 67, 96] and simplicity of the back-propagation learning algorithm [168], multi-layer feedforward neural networks have been frequently applied to systems modeling and identification [19, 52, 89, 145]. Chaotic dynamics presents good example for the neural network modeling, since chaos may provide us with rather complex target for learning. Despite this instinctive thought, modeling of chaotic dynamics can enhance the learning capability of neural networks. This is because:

(a) In general, chaotic dynamics is generated from a simple nonlinear dynamical system. Such simple nonlinear dynamical structure can be modeled by rather simple and small size neural networks.

(b) Due to the orbital instability, chaotic dynamics wanders among a variety of unstable periodic orbits embedded in the chaotic dynamical structure. Since chaotic wandering dynamics provides us with a rich information on its global dynamical structure, modeling of chaotic dynamics is much easier than other dynamics such as the periodic dynamics.

In fact, Lapedes and Farber [123] and Sato *et al.* [175] have reported successful results for modeling chaotic dynamics which even enhances the learning capability of neural networks. On the basis of their results, we study the learning capability of neural networks with time delayed feedbacks. Due to the existence of time delays, the neural network gives rise to infinite-dimensional dynamics. By the experimental studies of learning chaotic dynamics, we show that the infinite-dimensionality enhances the learning capability of the neural networks and demonstrate the significance of time delays in the synaptic connections of neural networks.

1.3.2 Identification of a parametrized family of chaotic dynamics

In **Chapter 3**, we extend the idea of modeling chaotic dynamics to the identification problem of a parametrized family of chaotic dynamics. Let us consider a chaotic dynamical system that exhibits a variety of chaotic time series with a change in the bifurcation parameters. The identification problem is to estimate the underlying bifurcation parameters of the chaotic time series without knowing the analytical structure of the parametrized family.

By using the technique of neural network modeling, an algorithm is presented for estimating the underlying bifurcation parameters. First, we construct "qualitatively similar" parametrized family of neural networks that model the sets of chaotic time series. "Qualitatively similar" parametrized family means that the family of neural networks exhibits "qualitatively similar" bifurcation phenomena as the original chaotic dynamics. Chaotic time series are then characterized in terms of the "qualitatively similar" parameters of the neural networks.

We emphasize that construction of the "qualitatively similar" parametrized family is possible since chaotic time series provides rich information on the global dynamical structure of the parametrized family. In this sense, chaotic dynamics enhances the neural network capability of identifying a parametrized family of dynamical systems and enables a novel application of the neural networks.

1.3.3 Detection of switch dynamics

In **Chapter 4**, we apply the system identification algorithm of Chapter 3 to the switch detection problem. Let us consider a chaotic time series generated from a dynamical system whose bifurcation parameters are occasionally switched. Then, the problem is to detect the switch dynamics in such switched chaotic time series without knowing the analytical structure of the parametrized family. First, the switched chaotic time series is divided into windows of short-term time series. From the set of windowed time series, "qualitatively similar" parametrized family of neural networks is constructed using the algorithm of Chapter 3. By characterizing the windows of short-term chaotic time series in terms of the "qualitative" parameters of neural networks, switch dynamics in the underlying bifurcation parameters can be detected.

1.3.4 Optimization by chaotic neural networks

In **Chapter 5**, we consider the optimization capability of neural networks in the presence of chaotic dynamics. Since the Hopfield-Tank experiment of applying neural networks to optimization problems [94], many researchers examined and modified the neural networks to solve optimization problems [13, 86, 210]. The well known problem of the optimization technique by the Hopfield-Tank neural network is the local minimum problem. The Hopfield-Tank network is known to have many local minimum solutions that are far from the optimum solution. The network is frequently trapped in such local minima depending on the choice of the initial condition. In order to escape from such local minima, stochastic dynamics is usually introduced to neural networks [2, 114]. Escape dynamics from the local minima can also be realized by introducing chaotic dynamics which destabilize the local minimum states.

According to Nozawa [154] who applied chaotic neural network [12] to solve the traveling salesman problem, remarkable optimization capability can be obtained when chaotic neural network exhibits chaotic behaviour. This observation is important, since it demonstrates that chaotic dynamics significantly improves the optimization capability of neural networks.

Motivated by the Nozawa experiment [154], we study global bifurcation structure of the chaotic neural networks applied to solve the traveling salesman problem. On the basis of the bifurcation study, global bifurcation scenario is presented for chaotic optimization dynamics. The scenario not only elucidates the dynamical mechanism of the chaotic optimization but it also provides a guideline for setting the system parameters for realizing an efficient chaotic optimization.

1.3.5 Simulated annealing in chaotic neural networks

In **Chapter 6**, we study algorithms for realizing *simulated annealing* [114] in chaotic neural network. Since the chaotic neural network studied in Chapter 5 gives rise to basically everlasting search dynamics for optimization problems, it is natural to introduce the idea of simulated annealing for gradually cooling down the chaotic dynamics to possibly optimum state.

Towards the simulated annealing in chaotic neural networks, chaotic simulated annealing (CSA) algorithm has been developed by Chen & Aihara [35]. In the CSA algorithm, the chaotic dynamics is harnessed by a cooling algorithm of a bifurcation parameter. Gradual cooling of the bifurcation parameter controls the chaotic search dynamics to converge to a stable equilibrium state with possibly optimum or near-optimum solution.

On the basis of the bifurcation scenario of Chapter 5, we show that the result of the chaotic simulated annealing algorithm is primarily dependent upon the global bifurcation structure of the chaotic neural networks. Unlike the stochastic simulated annealing [71, 114], *infinitely slow* chaotic annealing does not necessarily provide an optimum result. As the improved algorithms for CSA, adaptive CSA algorithm and learning algorithm are introduced for realizing simulated annealing in chaotic neural networks.

1.3.6 Chaotic dynamics for nonlinear function minimization problems

In **Chapter 7**, chaotic optimization technique of Chapters 5 and 6 are applied to the nonlinear function minimization problem. Again, global bifurcation scenario is presented for chaotic dynamical systems that solve the nonlinear optimization problem. The bifurcation scenario not only elucidates the dynamical mechanism of the chaotic search but it also shows that a simple simulated annealing algorithm for the chaotic dynamics does not necessarily provide an optimum result. Then, a learning algorithm is introduced for the chaotic optimization system to control the asymptotic measure of the chaotic search dynamics. Our numerical experiments show that the learning algorithm possibly works as "chaotic simulated annealing," which realizes gradual convergence of the chaotic search dynamics to possible optimum solution.

1.3.7 Conclusions, discussions, and future works

In **Chapter 8**, subject studies of Chapters 2-7 are summarized. Conclusive remarks are made to discuss functions of chaotic dynamics in neural networks applied to various engineering problems. Future works for thorough investigations of our subject studies are also provided.

Chapter 2

Back-propagation learning of infinite-dimensional dynamical systems

abstract: Back-propagation learning is introduced to recurrent neural network with time-delayed feedbacks (DRNN). Since the time-delays make the dynamics of the DRNN infinite-dimensional, the learning algorithm and the learning capability of the DRNN are different from those of the ordinary recurrent neural network (ORNN) with no time-delay. Two types of learning algorithms are developed for a class of DRNNs and they are tested for the learning of chaotic dynamics. Comparing the learning algorithms and the learning capability of DRNN with those of ORNN, functions of time delays in neural networks are considered.

2.1 Introduction

In neural physiology, various types of *time delays* such as the axon propagation delays and the synaptic transmission delays are experimentally observed. A natural direction of the neural network studies is to consider the functions of such time delays in neural systems. So far, many neural network models with time delays have been introduced and possible functions of the delays have been discussed. For instance, delayed synaptic connections are introduced to neural networks that solve a time-sequence recognition problem [19, 52, 95, 123, 145, 188]. In the network, the delayed synapses function for concentrating the input information in time and for recognizing the input time-sequence patterns. Such time-delayed neural networks are widely applied to practical engineering problems such as the speech recognition problem [19, 52, 95, 145, 188] and the nonlinear prediction problem [123]. Delayed feedback connections are also considered in a Hebbian-type associative memory neural network [39, 90, 113, 115, 184]. In the associative memory network, time delays destabilize the memory states of the Hebbian-type neural network and enable the network to sequentially recall the stored memories. Stability of the time-delayed neural networks has been also analyzed extensively [25, 30, 138, 156].

Although the neural network models with time delays have been mainly studied in the above contexts, the focus of the present Chapter is rather different from these studies. Our interest is in supervised learning of continuous-time recurrent neural network with time delayed feedbacks (DRNN). The supervised learning is to teach spatio-temporal dynamics to the DRNN by applying the back-propagation learning algorithm [167, 168]. Since the dynamics of the DRNN is described by retarded functional differential equations [82] whose dynamical class is different from the ordinary differential equational models of recurrent neural networks (ORNN), the learning algorithm and the learning capability of the DRNN are different from those of the conventional ORNN [48, 49, 77, 158, 163, 172, 173].

To the best of our knowledge, the supervised learning approach to DRNN has not yet been thoroughly investigated. Back-propagation learning is introduced to DRNN and the network capability of learning complex dynamics such as chaotic dynamics and irregular speech dynamics is examined in [195, 197]. By the dynamical analysis of DRNN, back-propagation learning algorithm for DRNN is briefly studied in [25] and a class of dynamical systems approximated by DRNNs is also discussed in [192].

The aim of the present Chapter is to consider possible functions of *time delays* in neural networks in the light of the supervised learning in DRNN. On the basis of the comparative studies which investigate advantages as well as disadvantages of DRNN over ORNN, functions of *time delays* in neural networks are studied.

The present Chapter is organized as follows. In Section 2.2, a standard mathematical model for DRNN is introduced and the learning algorithms for the DRNN are developed. Computational costs of the learning algorithms for DRNN and ORNN are compared. In Section 2.3, several numerical experiments are presented to show the learning capability of the DRNN. In Section 2.4, learning algorithm of the DRNN is further applied to speech modeling and synthesis. Section 2.5 is devoted for conclusions and discussions of the present Chapter.

2.2 Learning algorithms

2.2.1 Recurrent neural network with time delays

As a standard model for DRNN, let us consider the Kleinfeld model [115] which was introduced as a sequential memory generator:

$$\begin{aligned} \frac{d}{dt}x^i(t) &= -\frac{1}{R}x^i(t) + \sum_{j=1}^{N} T_{ij}V_j(t) + \sum_{j=1}^{N} D_{ij}V_{\mathcal{D}_j}(t) + \sum_{j=1}^{m} I_{ij}z_j(t), \\ & V_j(t) = G_j(x^j(t)), \;\; V_{\mathcal{D}_j}(t) = \int_{-\tau}^{0} V_j(t+s)\, \mathcal{D}_j(s)\, ds. \end{aligned} \tag{2.1}$$

As is shown in fig. 2.1, the network is composed of N-neurons, where the internal states, outputs, and delayed outputs of the i-th neuron are respectively described as $x^i(t)$, $V_i(t)$, and $V_{\mathcal{D}_i}(t)$ $(i = 1, .., N)$. The network also receives m-external inputs $z_i(t)$ $(i = 1, 2, .., m)$. The synaptic connections from the j-th neuron, the j-th delayed neuron, and the j-th external input to the i-th neuron are described by the weight matrices T_{ij}, D_{ij}, and I_{ij}, respectively. The delay function $\mathcal{D}_j(s)$ represents the response characteristics of the delayed synapse over the duration of τ. The delay function usually takes the form of the delta function, $\mathcal{D}_j(s) = \delta(s - \tau_j)$, the step function, $\mathcal{D}_j(s) = (\quad - s)/\tau_j$, or the exponential decay function, $\mathcal{D}_j(s) = \exp(-s/\tau_j)/\tau_j$. The input-output function $G_i(x)$ is represented by a monotonously-increasing function such as the sigmoidal function $G(x) = 2/(1 + \exp(-x)) - 1$. In case of no time-delay, i.e., $\tau = 0$, the network of eq. (2.1) becomes an ORNN widely used for the associative memory network [94] and the recurrent back-propagation network [48, 49, 77, 158, 163, 172].

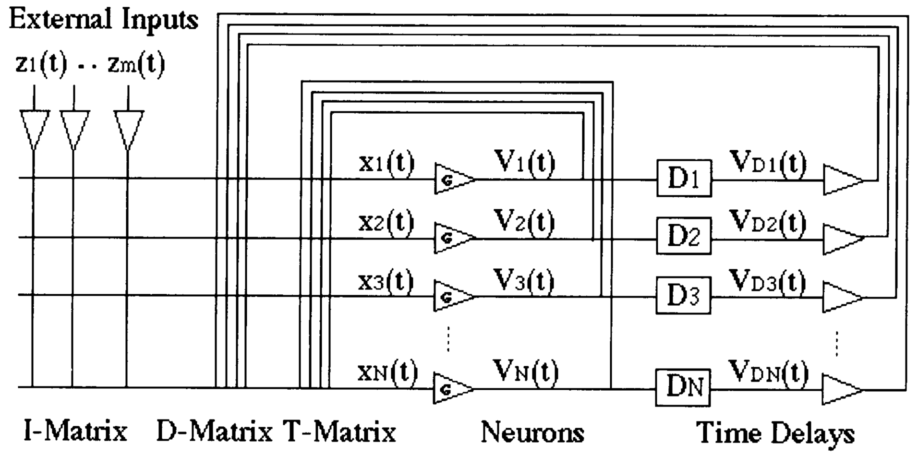


Figure 2.1: Schematic illustration of the Kleinfeld network of eq. (2.1). Dynamical states of the N-neurons $\{x^1, .., x^N\}$ are driven by the feedback connections from their outputs $\{V_1, .., V_N\}$ and the delayed outputs $\{V_{\mathcal{D}_1}, .., V_{\mathcal{D}_N}\}$. The network also receive m-external inputs $\{z_1, .., z_m\}$.

Eq. (2.1) can be described as a class of retarded functional differential equations (RFDEs) [82]:

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega), \tag{2.2}$$

where $\boldsymbol{x} = \{x^1, .., x^N\}$, $\boldsymbol{z} = \{z_1, .., z_m\}$, and $\boldsymbol{f} = \{f_1, .., f_N\}$. The dynamical system is infinite dimensional, since the state space $\boldsymbol{x}_t \in C([-\tau, 0], R^N)$ is represented by a continuous mapping of the interval $[-\tau, 0]$ into R^N according to $\boldsymbol{x}_t(\theta) = \boldsymbol{x}(t + \theta)$ for $\theta \in [-\tau, 0]$. The system dynamics is described by a function $\boldsymbol{f} : C([-\tau, 0], R^N) \times R^m \times R^K \to R^N$ with K-dimensional weight parameters $\Omega = \{T_{ij}, D_{ij}, I_{ij}\} \in R^K$.

In the next subsection, learning algorithms that adjust the weight parameters Ω are developed for a class of DRNNs described by eq. (2.2).

2.2.2 Real time recurrent learning algorithm

Let us consider the supervised learning to teach spatio-temporal dynamics to DRNN of eq. (2.2). The inverse problem for learning spatio-temporal dynamics can be stated as follows:

> "Classify the units of the DRNN into visible units $\{x^i | i \in \mathcal{V}\}$ and hidden units $\{x^i | i \notin \mathcal{V}\}$. Given an initial condition $\boldsymbol{x}_0$ and external inputs $\boldsymbol{z}(t)$, the DRNN gives rise to a unique solution $\boldsymbol{x}(t)$ satisfying $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega)$ for $t \in [0, \mathcal{T}]$. Then, find the adaptive parameters Ω that give rise to a solution $\boldsymbol{x}(t)$ approximately following a given target trajectory $\boldsymbol{\xi}(t) = \{\xi_i(t) | i \in \mathcal{V}\}$ as $x^i(t) \approx \xi_i(t)$ for $i \in \mathcal{V}$ and $t \in [-\tau, \mathcal{T}]$."

The back-propagation learning algorithm for the inverse problem can be formulated as follows. First, cost function for the adaptive parameters Ω is defined as

$$E(\Omega) = \int_0^{\mathcal{T}} dt \; \frac{1}{2} \sum_{i \in \mathcal{V}} \{x^i(t) - \xi_i(t)\}^2. \tag{2.3}$$

Then, the cost function of eq. (2.3) is minimized by the steepest descent method as follows:

$$\omega(n+1) = \omega(n) - \eta \frac{\partial E}{\partial \omega}(\Omega(n)) \quad \text{where} \quad \omega \in \Omega. \tag{2.4}$$

The first derivatives $\partial E / \partial \omega$ are computed as

$$\frac{\partial E}{\partial \omega} = \int_0^{\mathcal{T}} dt \; \sum_{i \in \mathcal{V}} \{x^i(t) - \xi_i(t)\} \frac{\partial x^i}{\partial \omega}(t), \tag{2.5}$$

where the values of $\partial x^i / \partial \omega$ in the right hand side are calculated by solving the first variational equations:

$$\begin{aligned} \frac{d}{dt} \frac{\partial x^i}{\partial \omega}(t) &= \sum_{j=1}^{N} \left[\int_{-\tau}^{0} ds \; \frac{\partial f_i}{\partial x_t^j(s)}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega) \frac{\partial x^j}{\partial \omega}(t+s) \right] \\ &\quad + \frac{\partial f_i}{\partial \omega}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega) \qquad (i = 1, 2, .., N) \end{aligned} \tag{2.6}$$

with an initial condition $\partial x^i / \partial \omega = 0$ for $t \in [-\tau, 0]$.

We remark that, in case of no time delay, i.e., $\tau = 0$, the present algorithm becomes the real-time recurrent learning (RTRL) algorithm [49, 173, 209] of the ORNN. Let us compare the computational cost of the RTRL algorithms for DRNN and ORNN. The main computational part of the RTRL algorithm is the numerical integration of the variational equations (2.6). In case of DRNN, the variational equations (2.6) are NK sets of RFDEs, while the equations (2.6) are NK sets of ordinary differential equations (ODEs) in case of ORNN. Compared to the ORNN, the RTRL algorithm for the DRNN is computationally costly, since the time delays make the variational equations (2.6) infinite-dimensional. The computational cost can be reduced by adopting the time-dependent recurrent learning (TDRL) algorithm [158, 172] to DRNN.

2.2.3 Time-dependent recurrent learning algorithm

Let us introduce the TDRL algorithm for DRNN. The TDRL algorithm computes the first derivatives $\partial E/\partial\omega$ by using the Lagrange multipliers $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, .., \lambda_N\}$ as follows.

First, we rewrite the cost function of eq. (2.3) as

$$\begin{aligned} L(\Omega) = \int_0^{\mathcal{T}} dt \Big[\sum_{i=1}^{N} \Big\{ \frac{1}{2}\, \delta_{i\in\mathcal{V}} \, \{x^i(t) - \xi_i(t)\}^2 \\ - \lambda_i(t)\, \{\dot{x}^i(t) - f_i(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega)\} \Big\} \Big] \end{aligned} \tag{2.7}$$

Then, the first derivatives $\partial L/\partial\omega$ are calculated as

$$\begin{aligned} \frac{\partial L}{\partial\omega} = \int_0^{\mathcal{T}} dt \quad & \Big[\sum_{i=1}^{N} \Big\{ \delta_{i\in\mathcal{V}} \, \{x^i(t) - \xi_i(t)\} \, \frac{\partial x^i}{\partial\omega}(t) \\ & + \lambda_i(t) \sum_{j=1}^{N} \int_{-\tau}^{0} ds \, \frac{\partial f_i}{\partial x_t^j(s)}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega) \, \frac{\partial x^j}{\partial\omega}(t+s) \\ & - \lambda_i(t) \, \frac{d}{dt}(\frac{\partial x^i}{\partial\omega}) + \lambda_i(t) \, \frac{\partial f_i}{\partial\omega}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega) \\ & - \frac{\lambda_i}{\partial\omega} \, \{\dot{x}^i(t) - f_i(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega)\} \Big\} \Big]. \end{aligned} \tag{2.8}$$

By the network dynamics of eq. (2.2), the final term of eq. (2.8) vanishes. Since the second term of the eq. (2.8) can be written by the transformation $t' = t + s$ as

$$\begin{aligned} & \int_0^{\mathcal{T}} dt \Big[\sum_{i=1}^{N} \Big\{ \lambda_i(t) \sum_{j=1}^{N} \int_{-\tau}^{0} ds \, \frac{\partial f_i}{\partial x_t^j(s)}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega) \, \frac{\partial x^j}{\partial\omega}(t+s) \Big\} \Big] \\ = & \int_0^{\mathcal{T}} dt' \Big[\sum_{i=1}^{N} \Big\{ \frac{\partial x^i}{\partial\omega}(t') \sum_{j=1}^{N} \int_{-\tau}^{0} ds \, \delta_{t'\in[0,\mathcal{T}+s]} \, \lambda_j(t'-s) \\ & \frac{\partial f_j}{\partial x_{t'-s}^i(s)}(\boldsymbol{x}_{t'-s}, \boldsymbol{z}(t'-s), \Omega) \Big\} \Big], \end{aligned} \tag{2.9}$$

the first derivatives $\partial L/\partial\omega$ become

$$\begin{aligned} \frac{\partial L}{\partial\omega} = \int_0^{\mathcal{T}} dt \quad & \Big[\sum_{i=1}^{N} \Big\{ \frac{\partial x^i}{\partial\omega}(t) \, \{ \, \delta_{i\in\mathcal{V}} \, (x^i(t) - \xi_i(t)) \\ & + \sum_{j=1}^{N} \int_{-\tau}^{0} ds \, \delta_{t\in[0,\mathcal{T}+s]} \, \lambda_j(t-s) \frac{\partial f_j}{\partial x_{t-s}^i(s)}(\boldsymbol{x}_{t-s}, \boldsymbol{z}(t-s), \Omega) \, \} \\ & - \lambda_i(t) \, \frac{d}{dt}(\frac{\partial x^i}{\partial\omega}) + \lambda_i(t) \, \frac{\partial f_i}{\partial\omega}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega) \Big\} \Big]. \end{aligned} \tag{2.10}$$

Suppose that the Lagrange multipliers $\boldsymbol{\lambda}(t)$ satisfy the following equations

$$\begin{aligned} \frac{d}{dt}\lambda_i(t) \; = \; & \delta_{i\in\mathcal{V}} \, (\xi_i(t) - x^i(t)) - \sum_{j=1}^{N} \int_{-\tau}^{0} ds \;\; \delta_{t\in[0,\mathcal{T}+s]} \, \lambda_j(t-s) \\ & \frac{\partial f_j}{\partial x_{t-s}^i(s)}(\boldsymbol{x}_{t-s}, \boldsymbol{z}(t-s), \Omega) \end{aligned} \tag{2.11}$$

with a boundary condition $\boldsymbol{\lambda}(\mathcal{T}) = 0$. By substituting eq. (2.11), the first, the second, and the third terms of eq. (2.10) become

$$\begin{aligned} & -\int_0^{\mathcal{T}} dt \sum_{i=1}^{N} \{ \frac{d}{dt}\lambda_i(t)\ \frac{\partial x^i}{\partial \omega}(t) + \lambda_i(t)\frac{d}{dt}\ \frac{\partial x^i}{\partial \omega}(t) \} \\ = & -\sum_{i=1}^{N} \{ \lambda_i(0)\ \frac{\partial x^i}{\partial \omega}(0) + \lambda_i(\mathcal{T})\ \frac{\partial x^i}{\partial \omega}(\mathcal{T}) \}. \end{aligned} \tag{2.12}$$

Since $\partial x^i/\partial\omega(0) = 0$ and $\boldsymbol{\lambda}(\mathcal{T}) = 0$, the above terms also vanish. Therefore, the first derivatives $\partial E/\partial\omega$ can be calculated by integrating the final term of eq. (2.10) by the following algorithm.

[Time-Dependent Recurrent Learning Algorithm]

(i) For a given initial condition $\boldsymbol{x}_0$ and external inputs $\boldsymbol{z}(t)$, solve the equations (2.2) forward in time. The solution curve $\boldsymbol{x}(t)$ $(t \in [0, \mathcal{T}])$ is stored in the computer memory.

(ii) For a boundary condition $\boldsymbol{\lambda}(\mathcal{T}) = 0$ and for target signal $\boldsymbol{\xi}(t)$, Lagrange multipliers $\boldsymbol{\lambda}(t)$ are calculated for $t \in [0, \mathcal{T}]$ by solving the equations (2.11) backward in time.

(iii) Using the Lagrange multipliers $\boldsymbol{\lambda}(t)$, the first derivatives are integrated as

$$\frac{\partial L}{\partial \omega} = \int_0^{\mathcal{T}} dt \sum_{i=1}^{N} \lambda_i(t)\ \frac{\partial f_i}{\partial \omega}(\boldsymbol{x}_t, \boldsymbol{z}(t), \Omega). \tag{2.13}$$

We remark that the same formula can be obtained by using the adjoint equations of the RFDE [82].

Although the memory storage of the entire solution curve $\boldsymbol{x}(t)$ $(t \in [0, \mathcal{T}])$ is required, the TDRL algorithm is computationally less expensive than the RTRL algorithm. While the RTRL algorithm requires to solve NK sets of RFDEs (2.6), the TDRL algorithm requires to solve only N sets of RFDEs (2.11). In Table 2.1, learning algorithms and their computational costs are summarized for DRNN and ORNN.

2.3 Numerical experiments

This Section presents experimental studies of applying the supervised learning to various types of DRNNs. As the teacher signals, periodic and chaotic signals are exploited. By comparing the learning capability of DRNN with that of ORNN, advantages of time-delays in recurrent neural network are studied.

In our numerical experiment, in order to avoid the numerical instability of integrating the network trajectory for long time interval, the cost function of eq. (2.3) is redefined for teacher signals $\boldsymbol{\xi}(t)$ divided into S-windows as

$$E(\Omega) = \sum_{n=1}^{S} \int_{(n-1)\mathcal{T}}^{n\mathcal{T}} dt \ \frac{1}{2} \sum_{i \in \mathcal{V}} \{x^i(t) - \xi_i(t)\}^2. \tag{2.14}$$

The modified cost function of eq. (2.14) is minimized by the quasi-Newton method:

$$\Omega(n+1) = \Omega(n) - H(\Omega(n)) \nabla E(\Omega(n)) \tag{2.15}$$

where $\nabla E(\Omega(n))$ and $H(\Omega(n))$ stand for a gradient vector and an approximate of the inverse Hessian of $E(\Omega(n))$. * The gradient vector $\nabla E(\Omega(n))$ is computed by the TDRL algorithm, where every differential equation is integrated by the Euler method. The Euler's integration algorithm for the RFDEs is described in detail in [53].

2.3.1 Figure eight

As the first learning example, figure-eight dynamics shown in fig. 2.2 (a) is employed for the teacher signal. As a DRNN, the following Kleinfeld model of eq. (2.1) with delta-function type delays is considered:

$$\frac{d}{dt} x^i(t) = -\frac{1}{R} x^i(t) + \sum_{j=1}^{N} T_{ij} V_j(t) + \sum_{j=1}^{N} D_{ij} V_j(t - \tau_j). \tag{2.16}$$

The network is composed of 2-units (x^1, x^2) $(N = 2)$, where the outputs are given by using the sigmoidal function as $V_i = G(x^i)$. The time constant and the delay parameters are fixed as $(R, \tau_1, \tau_2) = (10, 0.14, 0.07)$ and the teaching periods are set as $(S, \mathcal{T}) = (2, 3.0)$. The external inputs are not considered, i.e., $z(t) = 0$. The two variables (ξ_1, ξ_2) of the figure-eight dynamics of fig. 2.2 (a) are used as the teacher signals for the visible units (x^1, x^2), where no hidden unit is introduced.

Fig. 2.2 (b) shows the network dynamics after the 153-iterative learning processes. The network capability of reproducing quite similar dynamics to the teacher signal is recognized. In the studies of the supervised learning of the ORNN given by eq. (2.16) with $\tau = 0$, the figure-eight dynamics has been used as the benchmark test [158]. Due to the uniqueness of the solution in a continuous-time dynamical system, the figure-eight dynamics which has a crossing point at the origin can not be described by two dimensional dynamics and at least three dimensional dynamics is required. In order for ORNN with two visible units (x^1, x^2) to learn the figure-eight dynamics, at least one hidden unit is required. In fact, it has been reported that ten hidden units are required for ORNN to

* There is a variety of update formulas for estimating a series of $H(\Omega(n))$ using $E(\Omega(n))$ and $\nabla E(\Omega(n))$. In our numerical experiments, the Broyden-Fletcher-Goldfarb-Shanno formula with Luenberger's self-scaling formula [132] is exploited.

learn the figure-eight dynamics [158]. In contrast with the ORNN, the present experiment shows that no hidden unit is necessary for DRNN to learn the figure-eight dynamics. Since the time-delays make the DRNN infinite-dimensional, DRNN only with two visible units does not violate the uniqueness of the solution. This implies that, in case of the learning by DRNN, the dimension of the network dynamics can be increased by introducing the time-delays and that the time-delays can substitute a large number of hidden units.

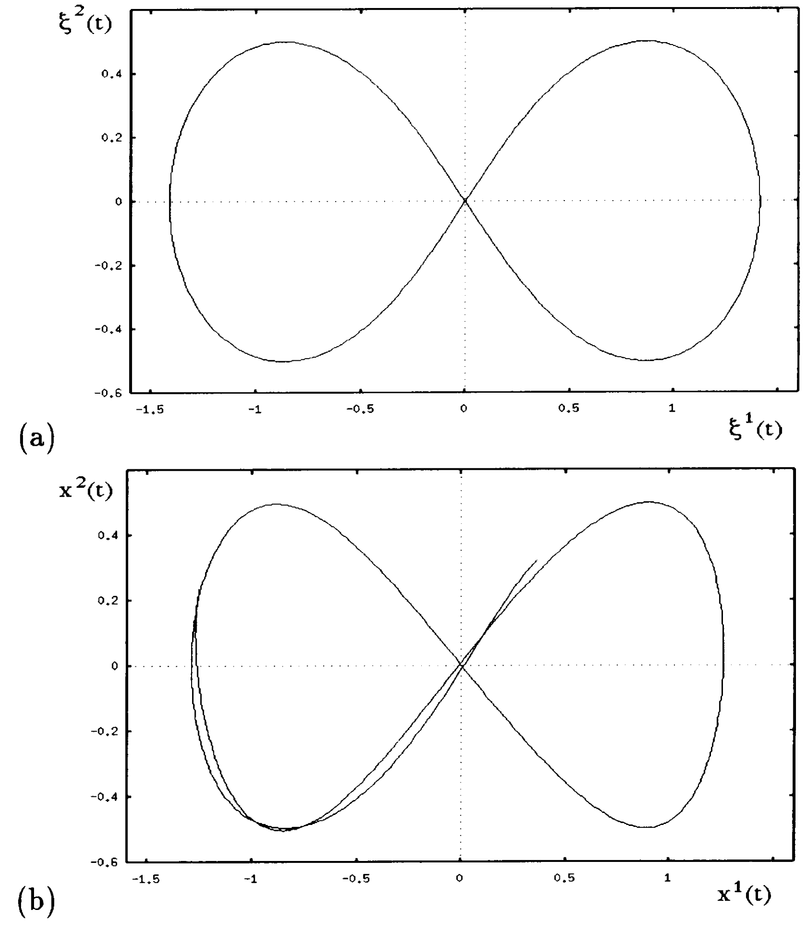


Figure 2.2: (a) Figure eight dynamics $(\xi_1(t), \xi_2(t))$ used for the first learning example. The dynamics has a crossing point at the origin. (b) Dynamics of the DRNN of eq. (2.16) in $(x^1(t), x^2(t))$-space after the 153-iterative learning. The network has two visible units, two delays, and no hidden unit.

2.3.2 Rössler equation

As the second example, the chaotic signal from the Rössler equations [165]:

$$\frac{d}{dt}\xi_1 = \xi_2 - \xi_3,$$

$$\begin{aligned} \frac{d}{dt}\xi_2 &= \xi_1 + 0.4\,\xi_2, \\ \frac{d}{dt}\xi_3 &= 2 - (4 - \xi_1)\,\xi_3, \end{aligned} \tag{2.17}$$

is employed as the teacher signal. Among the three variables, two variables (ξ_1, ξ_2) shown in fig. 2.4 (a) are used as the teacher signal.

In order for the learning of nonlinear chaotic dynamics, let us consider a DRNN having multi-layer-perceptron (MLP) structure:

$$\frac{d}{dt}x^i(t) = \sum_{k=1}^{h} H_{ik} G(\sum_{j=1}^{N} T_{kj} x^j(t) + \sum_{j=1}^{N} D_{kj} x^j(t - \tau_j) + I_k) \tag{2.18}$$

$$(i = 1, .., N)$$

As is shown in fig. 2.3, dynamical states of the N-neurons $\boldsymbol{x}(t) = \{x^1, .., x^N\}$ are driven by the outputs of the MLP. The MLP has one output layer composed of N-output units, one middle layer composed of h-middle units, and input layer which receives the inputs from the N-dynamical units $\{x^1(t), .., x^N(t)\}$ and their delayed outputs $\{x^1(t - \tau_1), .., x^N(t - \tau_N)\}$. The MLP structure is introduced to DRNN, since the MLP is known to have universal approximation capability [43, 67, 96].

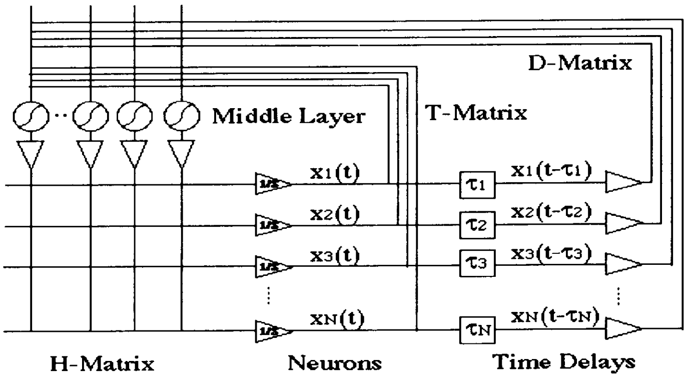


Figure 2.3: Schematic illustration of the MLP-type DRNN of eq. (2.18). Dynamical states of the N-neurons $\{x_1, .., x_N\}$ are driven by the outputs of the MLP, where the MLP has one output layer composed of N-output units, one middle layer composed of h-middle units, and input layer which receives the inputs from the dynamical units $\{x_1(t), .., x_N(t)\}$ and their delayed outputs $\{x_1(t - \tau_1), .., x_N(t - \tau_N)\}$.

For learning the Rössler dynamics of eq. (2.17), DRNN of eq. (2.18) with 2-visible units (x^1, x^2) which receive the teacher signals (ξ_1, ξ_2) is considered. No hidden unit is introduced for the DRNN. The delay parameters are set as $(\tau_0, \tau_1) = (0, 0.4)$, 5-middle units are located in the MLP $(h = 5)$, and the teaching periods are set as $(S, \mathcal{T}) = (100, 0.8)$. As the learning parameters, the weight parameters $\Omega = \{H_{ik}, T_{kj}, D_{kj}, I_k\}$ are adjusted by the learning algorithm.

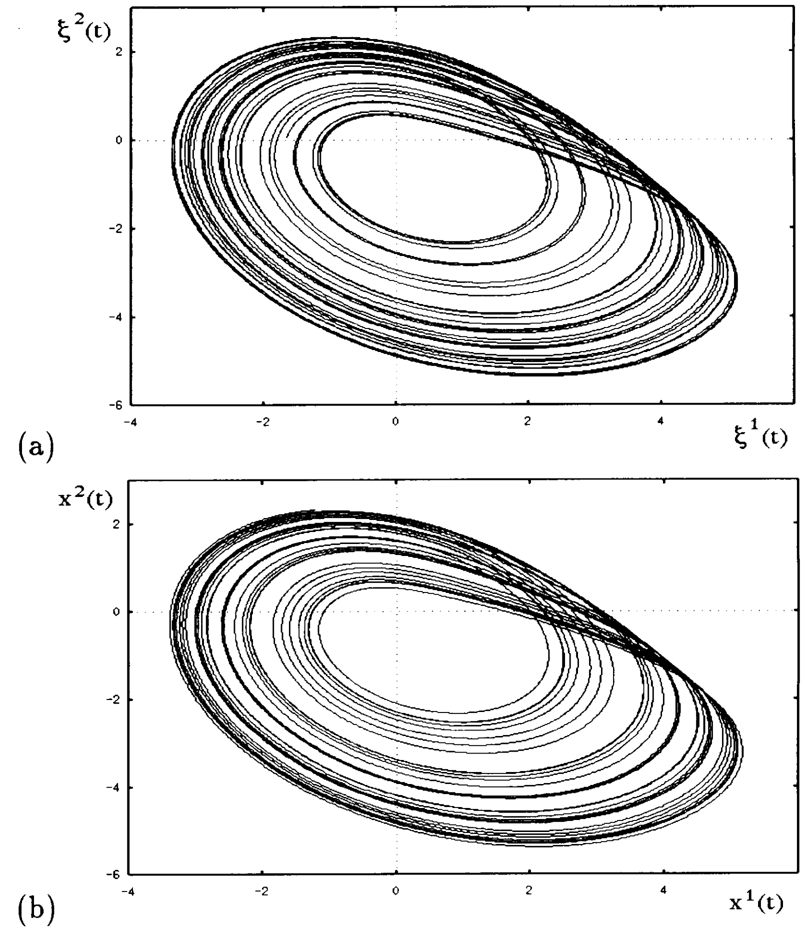


Figure 2.4: (a) Rössler dynamics of the eq. (2.17) in $(\xi_1(t), \xi_2(t))$-space. (b) Dynamics of the MLP-type DRNN of eq. (2.18) in $(x^1(t), x^2(t))$-space after 5000-iterative learning. The network has two visible units, one delay, and no hidden unit.

Fig. 2.4 (b) shows the network dynamics after the 5000-iterative learning processes. Reproduction of qualitatively similar chaotic dynamics is recognized. Quantitative similarity between the network dynamics and the target dynamics can be also confirmed by computing the Lyapunov dimension D_L [110], where the algorithm for computing the Lyapunov spectrum and the Lyapunov dimension of the RFDE is provided in [53]. The Lyapunov dimension is estimated as $D_L = 2.0216$ for the Rössler equations and $D_L = 2.0222$ for the DRNN.

Let us consider the case of learning chaotic dynamics by ORNN. In case of no time delay, i.e., $\tau = 0$, the DRNN of eq. (2.18) becomes the ORNN whose learning capability has been studied by several researchers [68, 175]. For such ORNN to learn chaotic dynamics from only two teacher signals (ξ_1, ξ_2), at least one hidden unit other than the two visible units is required. This is because the Peixoto theorem [160] states that chaotic dynamics can not be generated from continuous-time dynamical system with less than three degrees of freedom. In the experimental study of learning chaotic dynamics, ORNN with two visible units and one hidden unit is trained in [175]. Due to the chaotic property of sensitive dependence on initial condition, a slight error in initial condition of the hidden unit diverges in time and strongly perturbs the trajectory of the visible units. Hence, the initial condition of the hidden unit should be carefully determined. In the numerical studies of [174, 175], the initial condition of the hidden units is adjusted as the learning parameters and this seems to complicate the learning algorithm and the learning processes.

In contrast with the ORNN, the present experiment shows that the complicated learning of hidden units can be avoided by introducing time-delays in DRNN. The time-delays make the DRNN infinite-dimensional and realizes the learning of chaotic dynamics only by visible units.

2.3.3 Mackey-Glass equation

As the final teacher signal, the chaotic signal from the Mackey-Glass equation [137]:

$$\frac{d}{dt}\xi_1(t) = 0.2\,\frac{\xi_1(t-100)}{1+\xi_1{}^{10.0}(t-100)} - 0.1\xi_1(t), \tag{2.19}$$

is employed, where the single variable ξ_1 shown in fig. 2.5 (a) is used as the teacher signal.

Again, we train the MLP-type DRNN of eq. (2.18) with one visible unit x^1, one delay $x^1(\tau_1)$, and no hidden unit. The delay parameter, the number of the middle units, and the teaching periods are set as $\tau_1 = 100$, $h = 5$, $S = 100$, and $\mathcal{T} = 2.5$. By the learning algorithm, the weight parameters $\Omega = \{H_{ik}, T_{kj}, D_{kj}, I_k\}$ are adjusted.

Fig. 2.5 (b) shows the network dynamics after the 1450-iterative learning processes. Reproduction of qualitatively similar chaotic dynamics is recognized in figs. 2.5 (a) and (b). Quantitatively, a similar Lyapunov dimension can also be estimated as $D_L = 10.30$ for the original Mackey-Glass equation and $D_L = 10.14$ for the DRNN.

Let us consider the case of learning the Mackey-Glass dynamics by ORNN having no time-delay. In order to learn the Mackey-Glass dynamics which has a Lyapunov dimension of $D_L > 10$, one visible unit and more than ten hidden units are required. The present experiment shows that the DRNN can avoid introducing such large number of hidden units by making use of the time-delayed feedbacks.

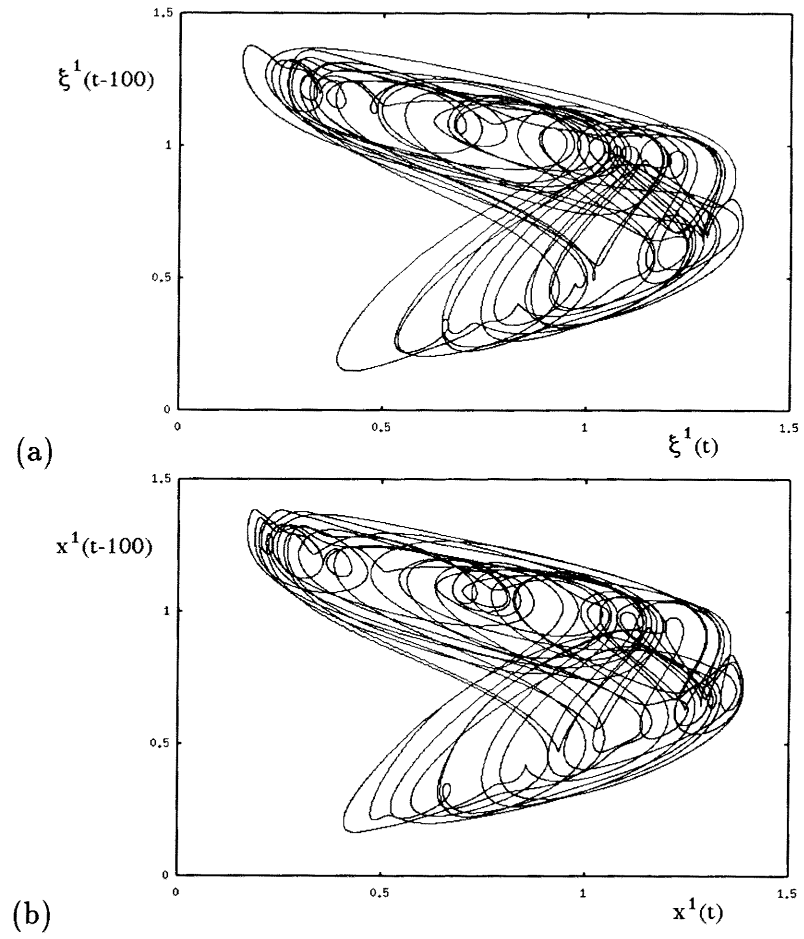


Figure 2.5: (a) Mackey-Glass dynamics of the eq. (2.19) in $(\xi_1(t), \xi_1(t-100))$-space. (b) Dynamics of the MLP-type DRNN of eq. (2.18) in $(x^1(t), x^1(t-100))$-space after 1450-iterative learning. The network has one visible unit, one delayed feedback, and no hidden unit.

2.4 Application to speech

2.4.1 Irregularity in speech

It has been reported that, under certain circumstances, human speech exhibits low-dimensional nonlinear dynamics. For example, pathological voices show evidence of low-dimensional chaos in Poincaré sections of the reconstructed speech trajectory [191]. Possible bifurcation route to chaos has been observed in non-stationary infant-cry [141] and chaotic dynamical systems analysis has been carried out for fricative consonants [152]. In normal phonation of vowels, irregularity in pitch-to-pitch variation has been also investigated in terms of low-dimensional nonlinear dynamics [26, 102, 122, 142, 143, 174, 195, 196, 197, 199]. Since irregular dynamical structure of the vowels is known to be quite important for natural perception of speech [47, 98, 116, 121], synthesis of irregular vowel dynamics by nonlinear neural network modeling is worthwhile investigating [174, 195].

This Section presents nonlinear modeling of such irregular vowel signal by the DRNN.

2.4.2 Nonlinear modeling of speech

In fig. 2.6, speech signal of the Japanese vowel /a/ (mausy003.ad)

$$\{\xi(t) : t = 1, 2, .., N_{data}\} \tag{2.20}$$

in the standard ATR (Advanced Telecommunications Research Institute International) data base [22] is shown. The signal is low-pass filtered with a cut-off frequency of 8 kHz and digitized with a sampling rate of 20 kHz and with 16 bit resolution. After removal of the initial transient phase and the final decay phase, the signal consists of 11 pitch periods of steady speech signal. Although the speech signal appears to be nearly periodic, each pitch waveform and the associated pitch period are different from each other.

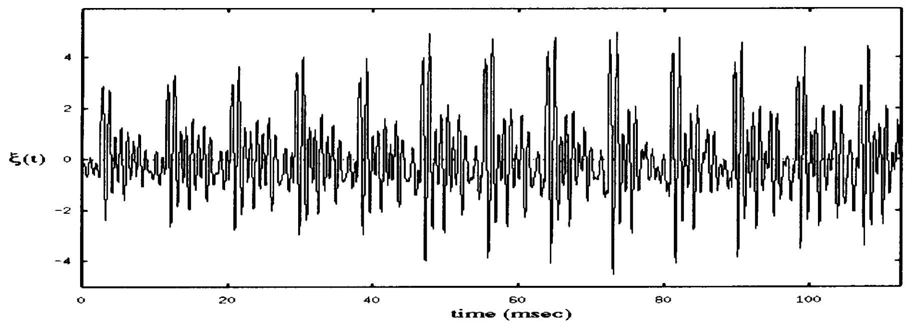


Figure 2.6: Speech signal of the Japanese vowel /a/ (mausy003.ad) taken from the ATR database. Consisting of 11 pitch periods, number of the data points is $N_{data} = 2200$.

Irregularity in the pitch-to-pitch variation can be clearly observed in the delay-coordinate vector space [186, 176]:

$$\boldsymbol{\xi}(t) = (\ \xi(t),\ \xi(t-\tau),\ \ldots,\ \xi(t-(d-1)\tau)\ )^T, \tag{2.21}$$

where T represents transposition and d and τ stand for the reconstruction dimension and the time lag, respectively. If the speech system that produces the vowel is deterministic and if the associated orbit is confined in a relatively low-dimensional attractor, the embedding theorems [186, 176] guarantee that topologically similar dynamics can be reconstructed in the delay-coordinate space. Figs. 2.7 show the trajectory of the speech signal reconstructed in 2- and 3-dimensional delay-coordinate spaces with $\tau = 3$. As is indicated by the arrows in figs. 2.7, behaviour of the reconstructed orbit is roughly summarized as follows:

1. The orbit spirals near to the origin.
2. As the orbit approaches the origin closely, it jumps upward and then bursts outside.
3. The trajectory spirals to the origin again.

The behaviour of the associated orbit resembles Shil'nikov's homoclinic chaos [179], implying that the irregularity in the pitch-to-pitch variation of the vowel might be caused by low-dimensional nonlinear dynamics.

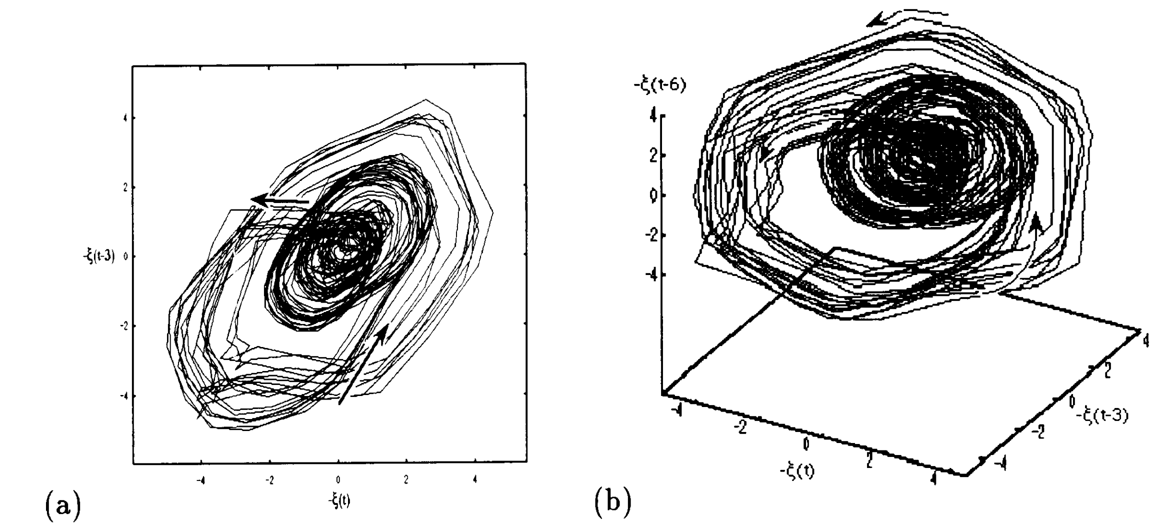


Figure 2.7: (a),(b): 2-dimensional and 3-dimensional dynamics of the speech signal of fig. 2.6, reconstructed in the delay-coordinate space $(-\xi(t), -\xi(t-3), -\xi(t-6))$. The arrows illustrate the system behaviour; as the orbit approaches the origin closely, it jumps upward and then bursts outside and again spirals to the center.

For nonlinear modeling of such irregular speech dynamics, an MLP-type recurrent neural network with multiple time delayed feedbacks:

$$\frac{d}{dt}x(t) = \sum_{k=1}^{h} H_k G(\sum_{j=0}^{K} D_{kj} x(t - j\tau) + I_k) \tag{2.22}$$

is exploited. As is shown in fig. 2.8, dynamical state of a single neuron $x(t)$ is driven by its time delayed feedbacks $x(t - j\tau)$ $(j = 0, 1, .., K)$.

The delay parameter, the number of the middle units of the MLP, the number of the delays, and the teaching periods are set as $\tau = 3.0$, $h = 40$, $K = 4$, $S = 20$, and $\mathcal{T} = 100$. By the learning algorithm, the weight parameters $\Omega = \{H_k, D_{kj}, I_k\}$ are adjusted.

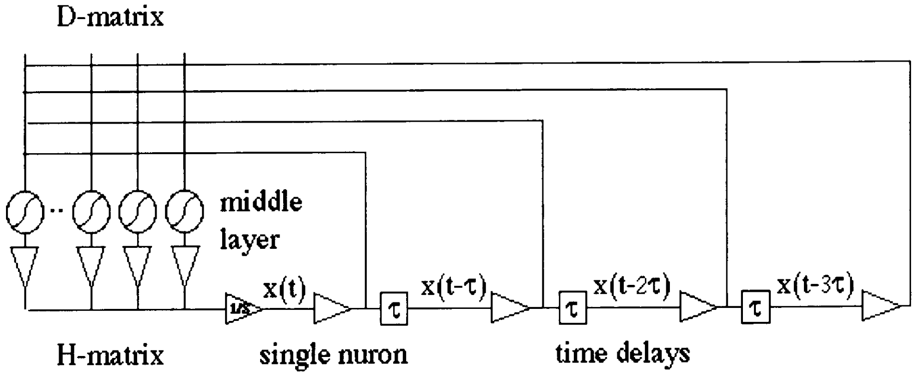


Figure 2.8: Schematic illustration of the DRNN of eq. (2.22). Dynamical state of a single neuron $x(t)$ is driven by the outputs of the MLP, where the MLP has one middle layer composed of h-middle units and input layer which receives the inputs from the delayed neural outputs $\{x(t), x(t-\tau), .., x(t-K\tau)\}$.

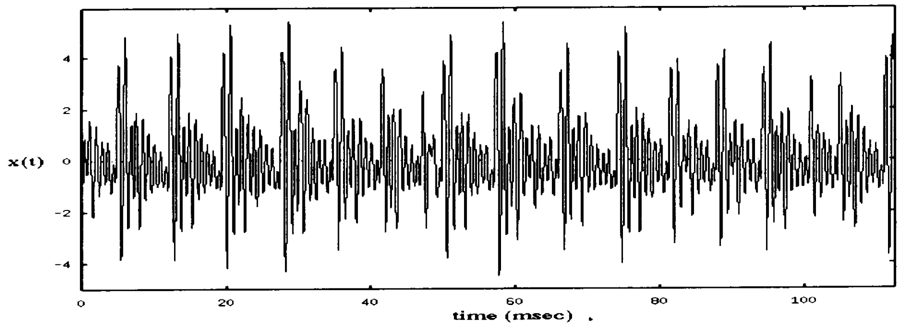


Figure 2.9: Speech signal synthesized by the DRNN of eq. (2.22).

Figs. 2.9 and 2.10 show the network dynamics after the 2000-iterative learning processes. Reproduction of qualitatively similar nonlinear dynamics is recognized in figs. 2.10 (a) and (b). Quantitatively, similar first Lyapunov exponent can be estimated as $\lambda_1 = 0.030$ for the DRNN and $\lambda_1 = 0.047$ for the real speech data. In order to compare long-term dynamical characteristic of the neural network with that of the real data, a subjective listening test was conducted to examine the sound quality of the D/A converted speech signal synthesized by the network. The signal sounded rather rough but definitely preserved its natural human speech quality. This demonstrates the network capability of reproducing the long-term characteristic of the natural speech sound.

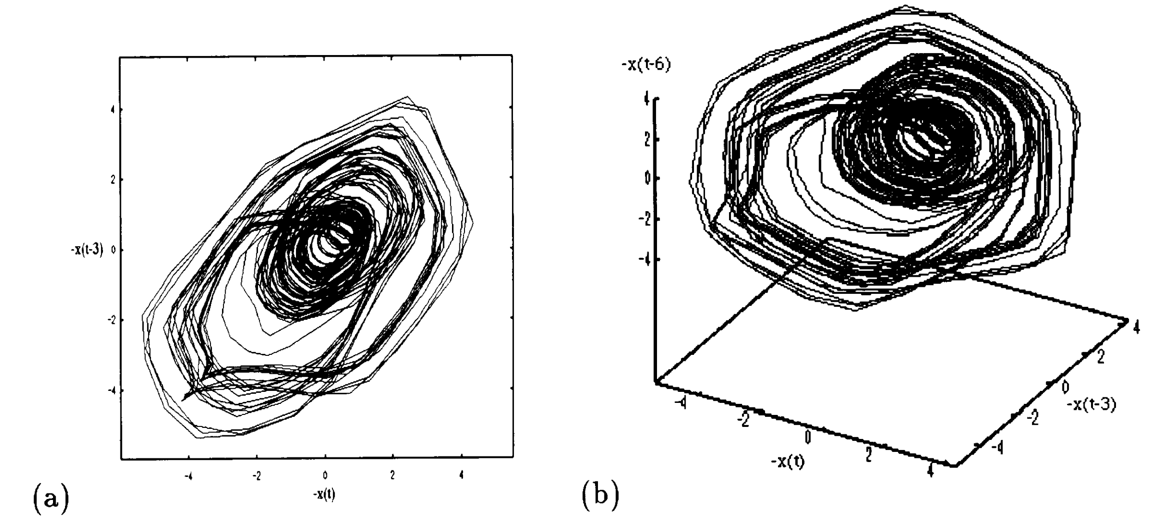


Figure 2.10: (a),(b): 2-dimensional and 3-dimensional dynamics of the DRNN of eq. (2.22) observed in the delay-coordinate space $(-x(t), -x(t-3), -x(t-6))$. with a time lag $\tau = 3$.

Let us consider the case of modeling the irregular speech dynamics without using time-delays. It has been reported by Sato *et al.* that an ORNN composed of a number of self-oscillating neurons [174] is capable of learning the complex speech dynamics. The network has a large number of hidden units that make the network size quite large and also complicate the learning processes. The present experiment shows that the irregular speech dynamics can be modeled by rather simple neural architecture using time-delayed feedbacks.

2.5 Conclusions and discussions

On the basis of the experimental studies that compare the learning algorithms and the learning capability of DRNN with those of ORNN, functions of *time delays* in neural networks are considered.

First, two learning algorithms, the RTRL algorithm and the TDRL algorithm, are introduced for general class of DRNNs. Compared to the RTRL algorithm for ORNN, the RTRL algorithm for DRNN is computationally costly due to the infinite-dimensionality of the DRNN. When real-time learning is not necessary, however, the learning cost can be largely reduced by adopting the TDRL algorithm to DRNN.

Second, several numerical experiments are carried out to compare the learning capability of the DRNN with that of the ORNN. In the supervised learning of the ORNN, the learning capability as well as the dimension of the ORNN can only be enhanced by increasing the number of the hidden units. Delicate learning is usually required for determining the initial condition of the hidden units [174, 175]. In contrast, the learning capability and the dimension of the DRNN can be enhanced simply by introducing the time-delays. By avoiding the complicated learning of the hidden units, DRNN only with a small number of visible units is shown to be capable of learning a variety of complex dynamics such as chaotic dynamics and irregular speech dynamics. In Table 2.1, advantages and disadvantages of the DRNN over the ORNN are summarized.

On the basis of the above discussions, we may consider that the time-delays in recurrent neural networks have a function of making the network dynamics infinite-dimensional and enhancing the network capability of learning higher-dimensional complex dynamics. In real neural systems, the time-delays may also have similar functions, although much more careful consideration and physiological understanding are indispensable.

In the present experiments, the learning capability of recurrent neural networks with only delta-function type delays are studied. As is described in Section 2.2, various types of delays can be considered in the neural network and it may be interesting to study the learning capability of the DRNN with other type of delays. It is also a worthwhile investigation to include the delay function parameters into the learning parameters and adjust the delays during the learning process.

It is also possible to discuss the functions of delays in the light of the class of dynamics approximated by neural networks. Although the approximation capability of the ORNNs has been studied in [68], approximation capability of the DRNN has not yet been thoroughly investigated. We have a preliminary result on the approximation capability of a particular class of DRNNs and the result will be reported elsewhere [192].

Finally, delayed feedback systems described by RFDEs can be found in many engineering systems such as a passive optical resonator [99]. This laser system is known to exhibit rich dynamical phenomena such as higher-harmonic bifurcations that give rise to multi-stability of infinitely many periodic attractors. This multi-stability has been actually used as a memory device for complex information coding [4]. This implies that the neural network with time-delays may also give rise to this type of multi-stability and might be capable of learning and embedding many attractors in the network dynamics. Learning multiple dynamics may also provide us with an interesting new applications of the DRNN.

Table 2.1: Learning algorithm and learning capability for DRNN and ORNN.

	DRNN	*ORNN*
RTRL Algorithm	Solving NK Sets of RFDEs	Solving NK Sets of ODEs
TDRL Algorithm	Solving N Sets of RFDEs	Solving N Sets of ODEs
Enhancement of Learning Capability	Increasing Time-Delays	Increasing Hidden Units

Chapter 3

Recognizing chaotic time series

abstract: Consider a chaotic dynamical system that exhibits a variety of chaotic time series with a change in the bifurcation parameters. This Chapter presents an algorithm for estimating the underlying bifurcation parameters of the chaotic time series in experimental situation in which no *a priori* analytical knowledge of the dynamical system is available. First, we construct "qualitatively similar" parametrized family of neural networks that model the chaotic time series. "Qualitatively similar" parametrized family means that the family of neural networks exhibits "qualitatively similar" bifurcation phenomena as the original. Chaotic time series are then characterized in terms of the "qualitatively similar" bifurcation parameters of the neural networks. We call the characterization of chaotic time series in terms of the underlying bifurcation parameters "chaotic time series recognition." Several numerical experiments using the Rössler equations show efficiency of the algorithm. The effect of observational noise included in chaotic time series is also considered.

3.1 Introduction

Suppose we have a chaotic dynamical system that exhibits a variety of oscillatory patterns in accordance with its external conditions, i.e., bifurcation parameters. Suppose also that several chaotic time series can be recorded at different bifurcation parameter values while we have no priori knowledge of the dynamical system, namely, the functional form of the dynamical system, its dependence of the bifurcation parameters, and the bifurcation parameter values which give rise to the recorded time series. Under these conditions, we consider a systematic characterization of the chaotic time series.

Various techniques have been developed in recent years to characterize chaotic time series [1, 34, 101, 109, 189, 208]. The algorithms estimate statistical properties of the underlying chaotic attractors such as fractal dimension [78, 105], Lyapunov spectrum [51, 171, 212], and Kolomogorov-Sinai entropy [57]. Our approach is entirely different from these techniques. Since the variety of oscillatory patterns is controlled by bifurcation parameters, it is natural to characterize the chaotic time series in terms of the associated bifurcation parameters. In order to distinguish our approach from conventional ones, we call the problem of estimating the underlying bifurcation parameters of chaotic time series "chaotic time series recognition."

"Chaotic time series recognition" can be directly applied to actual engineering problems, e.g., recognition of oscillatory states of chaotic chemical sensor [169]. Analytical understanding of the dynamics of complicated chemical sensor system is almost always impossible. With a change in the external conditions such as chemical ingredients, the system exhibits bifurcation phenomena with a variety of chaotic time series. In this case, it would be more efficient to characterize the chaotic time series in terms of the bifurcation parameter values rather than the statistical properties of the chaotic attractors. If the estimation of the underlying bifurcation parameters is possible, the chaotic oscillatory patterns can be classified precisely and one can "measure" the external conditions just by observing the output of the sensor.

Estimation of the exact bifurcation parameter values only from chaotic time series is practically impossible, since the problem provides no information about the underlying bifurcation parameters. However, it is possible to estimate "qualitatively similar" bifurcation parameter values. Here, "qualitatively similar" bifurcation parameters mean that the parameters give rise to a family of dynamical systems which exhibits bifurcation phenomena qualitatively similar to the originals. In other words, the chaotic time series can be recognized based on a parametrized family of dynamical systems qualitatively similar to the original.

In the area of nonlinear prediction techniques [32, 41, 54, 123, 139, 182], several interesting studies have been done for reconstructing a qualitatively similar parametrized family of dynamical systems using a parametrized family of nonlinear predictors. In [41], although no specific result is presented, problems for reconstructing a parametrized family of dynamical systems using a parametrized family of nonlinear predictors are discussed. In [32], the experimental result of reconstructing a one-parameter family of discrete-time dynamical systems, assuming that the sets of the bifurcation parameter values are all known as well as the associated time series, is reported.

Our approach to the problem is based on a simple algorithm proposed in [198]. The algorithm attempts to reconstruct a family of discrete-time dynamical systems from chaotic time series under the condition that the underlying bifurcation parameter values are not known. The efficiency of the algorithm is shown by reconstructing two families of discrete-time dynamical systems: the Hénon family with two parameters and the coupled logistic/delayed-logistic family with three parameters.

For actual application in constructing qualitatively similar parametrized family of nonlinear predictors for real world systems, however, the algorithm should be applicable to the family of continuous-time dynamical systems. The algorithm should also be robust against the observational noise which surely exists in laboratory experiments. Therefore, this Chapter gives a modified algorithm so that one can construct a qualitatively similar family of continuous-time dynamical systems only from chaotic time series including observational noise. Using the algorithm, the applicability of the reconstructed family of nonlinear predictors to the problem for "chaotic time series recognition" is studied.

This Chapter is organized as follows. Section 3.2 illustrates an algorithm for reconstructing an unknown parametrized family of continuous-time dynamical systems. In Section 3.3, the algorithm is tested against the Rössler family with two parameters. In Section 3.4, we make an experimental study on recognizing chaotic time series based on a reconstructed family for the Rössler equations. Section 3.5 is devoted to discussions of our future problems.

3.2 Reconstructing a parametrized family of continuous-time dynamical systems

Consider a parametrized family of continuous-time dynamical systems:

$$\frac{d\eta_t}{dt} = f(p, \eta_t), \quad \eta_t \in R^D, \quad p \in R^m \tag{3.1}$$

and their observations

$$\{\xi_t(p) = g(\eta_t(p)) : 0 \leq t \leq T\} \tag{3.2}$$

at I different sets of parameter values

$$p \in \{p(i)\}_{i=1,2,..,I} \tag{3.3}$$

where $g : R^n \to R^1$ is a smooth observation function and $\eta_t(p)$ is a solution for eq. (3.1) at the parameter values p. Here we assume the following:

(i) The functional form of the parametrized family of vector fields $f : R^m \times R^D \to R^D$ is not known; f is assumed to be smooth.

(ii) The functional form of g is not known; g is assumed to be smooth.

(iii) $m = \dim p$ and the sets of parameter values $\{p(i)\}_{i=1,..,I}$ are not known; $\{p(i)\}_{i=1,..,I}$ are assumed to be closely located in the parameter space.

(vi) $D = \dim \eta_t$ is not known.

(v) I sets of observed time series $\{\xi_t(p(i)) : 0 \leq t \leq T\}$ are all chaotic.

Under the conditions (i)-(vi), we reconstruct an unknown parametrized family of continuous-time dynamical systems only from the chaotic time series. The algorithm is composed of three steps. First, observational noise of the measured time series is reduced by an averaging filter, and qualitatively similar orbits are reconstructed using a delay-coordinate transformation. Second, nonlinear predictors which forecast the chaotic time series are sought using a same parametrized family of neural networks. Third, effective bifurcation parameters are extracted from the many parameters of the nonlinear predictors by principal component analysis.

3.2.1 Average filtering of chaotic time series

Since time series are usually sampled digitally in laboratory experiments, let eq. (3.2) be rewritten as

$$\{\xi_n(p) = g(\eta_{n\Delta t}(p)) : n = 1, 2, .., N\} \tag{3.4}$$

where $\eta_t(p)$ is a solution for eq. (3.1) at the parameter values p and Δt is a sampling rate. We assume that the sampling rate Δt is known.

In order to reduce the observational noise effect, an averaging filter is applied to the time series of eq. (3.4). The filtered time series are given by

$$\begin{aligned} \{\hat{\xi}_n(p(i)) &= \frac{\sum_{k=n}^{n+W} \xi_k(p(i))}{W\Xi} : n = 1, 2, .., N - W\}_{i=1,2,..,I} & (3.5) \\ \text{with} \quad \Xi &= \max_{n,i}\{|\sum_{k=n}^{n+W} \xi_k(p(i))|\} & (3.6) \end{aligned}$$

where W is the window length of the moving average and Ξ is a normalization constant.

3.2.2 Nonlinear predictors

From the filtered time series $\{\hat{\xi}_n(p(i)) : n = 1, 2, .., N - W\}_{i=1,2,..,I}$, a d-dimensional trajectory $\{X_n(p(i)) : n = (d-1)\tau + 1, .., N - W\}_{i=1,2,..,I}$ is reconstructed by using a delay-coordinate [186, 176]:

$$\begin{aligned} X_n(p) &= {}^T({}^1x_n(p), {}^2x_n(p), ..., {}^dx_n(p)) \\ &= {}^T(\hat{\xi}_n(p), \hat{\xi}_{n-\tau}(p), \hat{\xi}_{n-2\tau}(p), .., \hat{\xi}_{n-(d-1)\tau}(p)) \end{aligned} \tag{3.7}$$

where T denotes transposition. The Filtered Delay Embedding Prevalence Theorem [176] guarantees that the reconstructed trajectory $\{X_n(p(i))\}_{i=1,2,..,I}$ is qualitatively the same as the original $\{\eta_{n\Delta t}(p(i))\}_{i=1,2,..,I}$.

Next, for each reconstructed trajectory, $\{X_n(p(i))\}_{i=1,2,..,I}$, we seek an ordinary differential equation:

$$\frac{d\phi^t}{dt} = F(\Omega, \phi^t) \tag{3.8}$$

that satisfies

$$\begin{aligned} X_{n+k}(p(i)) &= \phi^{k\Delta t}\left(\Omega(p(i)), X_n(p(i))\right) \\ \text{for } k &= 1, 2, .., K \end{aligned} \tag{3.9}$$

where $\Omega \in R^L$ stands for a set of parameters of nonlinear function $F(\cdot,\cdot)$, $\phi^t : R^L \times R^d \to R^d$ stands for a solution of the eq. (3.8) at $\Omega = \Omega(p(i))$ with an initial condition $\phi^0(\Omega, X) = X$.

As the nonlinear function model $F : R^L \times R^d \rightarrow R^d$, multi-layer perceptron (MLP) is exploited, where MLP composed of three-layers (d-units in the input layer, d-units in the output layer, and h-units in the hidden layer) is given by

$$F(\Omega, \phi) = {}^T(f_1(\Omega,\phi), f_2(\Omega,\phi), .., f_d(\Omega,\phi)) \tag{3.10}$$

where

$$\begin{aligned} f_k(\Omega,\phi) &= \sum_{j=1}^{h} \omega_{(k-1)h+j}\ \sigma(\sum_{i=1}^{d} \omega_{dh+(j-1)d+i}\ {}^i\phi + \omega_{2dh+j}\ )\ \ (k = 1,2,..,d), \\ \sigma(x) &= \frac{2.0}{1+e^{-x}} - 1.0, \\ \Omega &= {}^T(\omega_1, \omega_2, .., \omega_L)\ \text{ with }\ L = (2d+1)h, \\ \phi &= {}^T(\ {}^1\phi,\ {}^2\phi, ..,\ {}^d\phi\ ). \end{aligned}$$

The parameters $\{\Omega(p(i))\}_{i=1,2,..,I}$ which give rise to the reconstructed dynamics $\{X_n(p(i))\}_{i=1,2,..,I}$ are computed in the following manner. First, the reconstructed orbits are periodically ordered as

$$\{X_n(p(1))\}, \{X_n(p(2))\}, .., \{X_n(p(I))\}, \{X_n(p(I+1))\}(= \{X_n(p(1))\}), .. \tag{3.11}$$

Second, $\Omega(p(1))$ which minimizes the cost function

$$U(\Omega) = \sum_{n=(d-1)\tau+1}^{N-W-K} \sum_{k=1}^{K} \frac{1}{2} \mid X_{n+k}(p(1)) - \phi_{\cdot}^{k\Delta t}(\Omega, X_n(p(1))) \mid^2 \tag{3.12}$$

is computed via the quasi-Newton method * [132] (see Appendix A-1) where the initial condition for $\Omega(p(1))$ is given by a set of random values uniformly distributed over $[0, 0.1]^L$. In a similar manner, $\Omega(p(i))$ $(2 \leq i)$ is computed by minimizing the cost function (3.12) defined for the reconstructed trajectory $\{X_n(p(i))\}$, where $\Omega(p(i-1))$ is used as the initial condition instead of random values. In our numerical experiments, we set the number of iterations of the minimization procedures of the quasi-Newton method to 20 for $i < 10I$ and 40 for $10I \leq i$.

The procedures for computing $\{\Omega(p(i))\}_{i=1,2,..}$ are repeated until they converge to a periodic sequence as

$$\Omega(p(N_I)), \Omega(p(N_I+1)), .., \Omega(p(N_I+I)), \Omega(p(N_I+I+1))(= \Omega(p(N_I))), .. \tag{3.13}$$

where N_I is assumed to be sufficiently large.

* In the quasi-Newton method, a local minimum of the cost function (3.12) is sought by the iterative procedure of $\Omega_{n+1} = \Omega_n - H_n \nabla U(\Omega_n)$ where $\nabla U(\Omega_n)$ and H_n stand for a gradient vector and an approximate of the inverse Hessian of $U(\Omega)$ at $\Omega = \Omega_n$. There is a variety of update formulas for estimating a series of $\{H_n\}$. In our numerical experiments, we exploit the Broyden-Fletcher-Goldfarb-Shanno formula with Luenberger's self-scaling formula.

3.2.3 Extracting principal bifurcation parameters

In the final step of our algorithm, principal component parameters are extracted from the L-dimensional parameters of Ω by the Karuhnen-Loéve (KL) transform [14, 21, 28, 140]. First, we consider the subsequence of the parameters $\{\Omega(p(i)) : i = N_I, N_I + 1, .., N_I + N_J - 1\}$ and compute $\{\delta\Omega_i : i = 1, 2, .., N_J\}$ and Ω_0 as

$$\begin{aligned} \Omega_0 &= \frac{1}{N_J}\sum_{i=1}^{N_J}\Omega(p(N_I + i - 1)), & (3.14)\\ \delta\Omega_i &= \Omega(p(N_I + i - 1)) - \Omega_0, & (3.15) \end{aligned}$$

where N_J stands for the number of the elements. Second, the multivariate distribution of $\{\delta\Omega_i : i = 1, 2, .., N_J\}$ is computed in terms of the covariance matrix:

$$\Omega_{L\times L} = \frac{1}{N_J}\sum_{i=1}^{N_J} \delta\Omega_i \ {}^T\delta\Omega_i. \tag{3.16}$$

Since $\Omega_{L\times L}$ has non-negative eigenvalues $\{\lambda_1, \lambda_2, .., \lambda_L\}$, we arrange them in descending order

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0. \tag{3.17}$$

Applying KL-transformation to $\delta\Omega$, the principal parameters are given by

$$\Gamma = (\gamma_1, \gamma_2, .., \gamma_L) = {}^T[u_1 \mid u_2 \mid \cdots \mid u_L]^{-1}\ \delta\Omega \tag{3.18}$$

where $\{u_1, u_2, .., u_L\}$ stand for the eigenvectors corresponding to $\{\lambda_1, \lambda_2, .., \lambda_L\}$.

Since transformation (3.18) diagonalizes the covariance matrix (3.16) in Γ-space, the diagonal elements $\{\lambda_1, \lambda_2, .., \lambda_L\}$ represent the significance of their corresponding principal parameters $\{\gamma_1, \gamma_2, .., \gamma_L\}$. Finally, by computing the normalized eigenvalues

$$\Lambda_i = 100 \times \frac{\lambda_i}{\sum_{j=1}^{L}\lambda_j}\ [\%] \quad (i = 1, 2, .., L), \tag{3.19}$$

the number of significant parameters M is determined. If the sample parameters $\{\delta\Omega_i : i = 1, 2, .., N_J\}$ are all confined in the m-dimensional linear subspace of $\delta\Omega$, one can expect that $M = m$ and that $\Gamma_m = {}^T(\gamma_1, \gamma_2, .., \gamma_m)$ represents the significant set of parameters for the nonlinear predictors (3.8).

With respect to the significant parameters Γ_m, the m-parameter family of nonlinear predictors is given by

$$\frac{d\phi^t}{dt} = F(\Omega(\Gamma_m), \phi^t) \tag{3.20}$$

where

$$\Omega(\Gamma_m) = {}^T[u_1 \mid u_2 \mid \cdots \mid u_L]^{-1}[\Gamma_m \mid 0] + \Omega_0 \tag{3.21}$$

where 0 denotes an $(L-m)$-dimensional 0 column vector.

Since the sequence of the significant parameters

$$\Gamma_m(p(1)) \to \Gamma_m(p(2)) \to \cdots \to \Gamma_m(p(N_J)) \tag{3.22}$$

can be considered as the image of the sequence of the original bifurcation parameters

$$p(1) \to p(2) \to \cdots \to p(N) \to p(I+1)(=p(1)) \to \cdots \tag{3.23}$$

in the significant parameter space of nonlinear predictors, we call the original sequence of eq. (3.23) "bifurcation path" and its corresponding sequence of eq. (3.22) "bifurcation locus."

3.3 Numerical experiment on Rössler equation

Let us test our algorithm against the Rössler equations with two parameters:

$$\begin{aligned} \frac{d\,{}^1\eta_t}{dt} &= {}^2\eta_t - {}^3\eta_t \\ \frac{d\,{}^2\eta_t}{dt} &= {}^1\eta_t - p_2\,{}^2\eta_t \\ \frac{d\,{}^3\eta_t}{dt} &= p_3\,{}^1\eta_t - p_1\,{}^3\eta_t + {}^1\eta_t\,{}^3\eta_t. \end{aligned} \tag{3.24}$$

We selected the Rössler equations as our example for two reasons; The bifurcation structures have been well studied (e.g., [70]) and the bifurcation structure is fairly moderate.

In this experiment, p_3 is fixed as

$$p_3 = 0.3 \tag{3.25}$$

so that the eqs. (3.24) is considered as a two-parameter family. Fig. 3.1 (a) shows a local bifurcation diagram of the eq. (3.24) with

$$(p_1, p_2) \in [\,4.3,\ 5.7\,] \times [\,0.30,\ 0.36\,], \tag{3.26}$$

while fig. 3.1 (b) shows a global bifurcation diagram with

$$(p_1, p_2) \in [\,3.814,\ 8.814\,] \times [\,0.25,\ 0.429\,]. \tag{3.27}$$

The bifurcation diagram shows the existence domain of periodic attractors. The color indicates the number of their periods: Blue indicates 1 or 8 or 15, red indicates 2 or 9 or 16, purple indicates 3 or 10 or 17, green indicates 4 or 11 or 18, sky indicates 5 or 12 or 19, yellow indicates 6 or 13 or 20, white indicates 7 or 14, and black indicates non-periodic attractor (quasi-periodic or chaotic or diverge to infinity) or periodic attractor whose period is greater than 21.

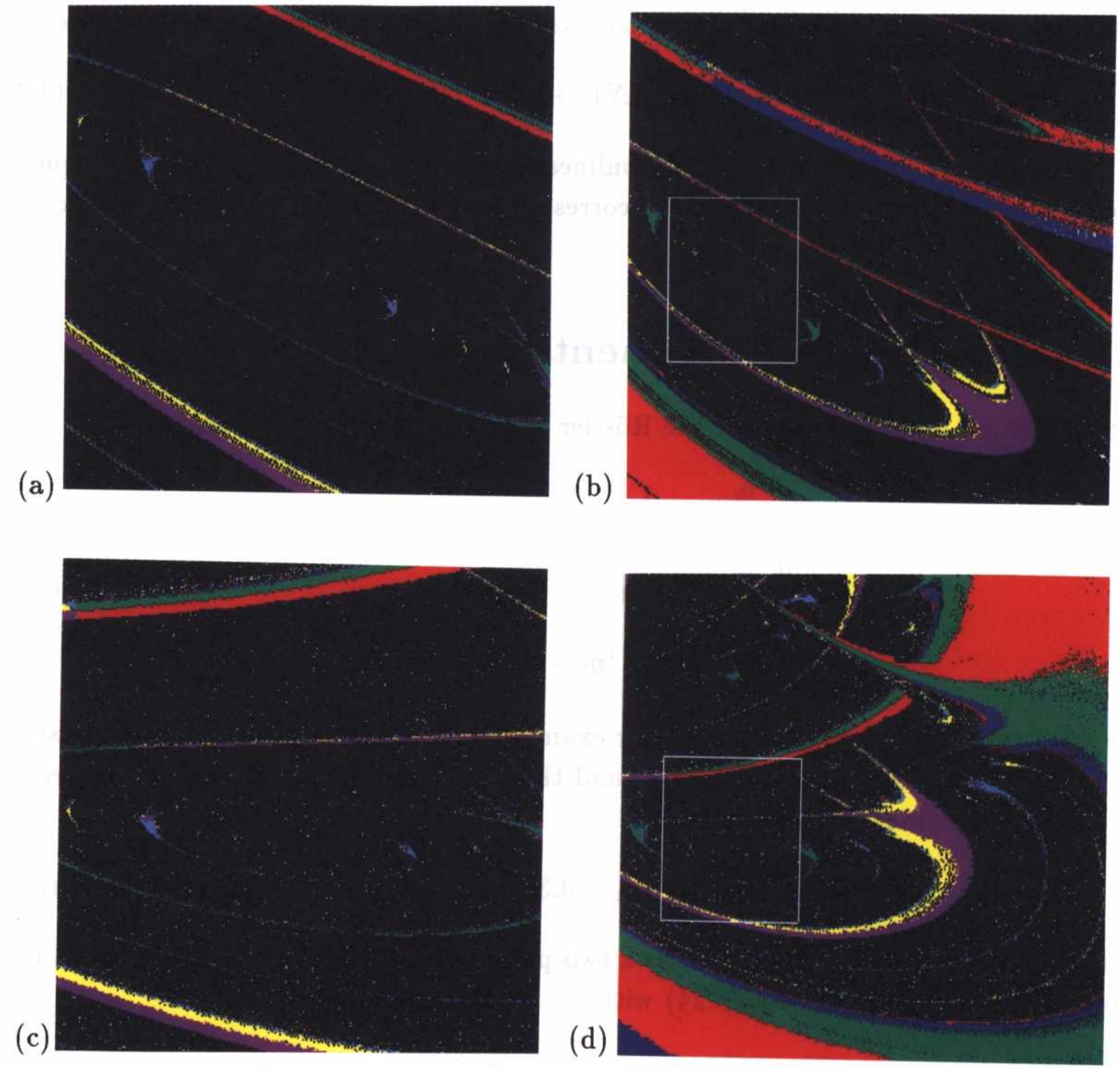


Figure 3.1: (a) (p_1, p_2)-bifurcation diagram of the Rössler equations with $(p_1, p_2) \in [4.3, 5.7] \times [0.30, 0.36]$. (b) (p_1, p_2)-bifurcation diagram of the Rössler equations with $(p_1, p_2) \in [3.814, 8.814] \times [0.25, 0.429]$. The rectangle region corresponds to the bifurcation diagram of fig. 3.1 (a). (c) (γ_1, γ_2)-bifurcation diagram of the nonlinear predictors (3.20) reconstructed in subsection 3.3.1 with $(\gamma_1, \gamma_2) \in [-0.103, 0.105] \times [-0.044, 0.051]$. (d) $(\gamma_1, -\gamma_2)$-bifurcation diagram of the nonlinear predictors (3.20) reconstructed in subsection 3.3.1 with $(\gamma_1, \gamma_2) \in [-0.175, 0.568] \times [-0.123, 0.160]$. The rectangle region corresponds to the bifurcation diagram of fig. 3.1 (c).

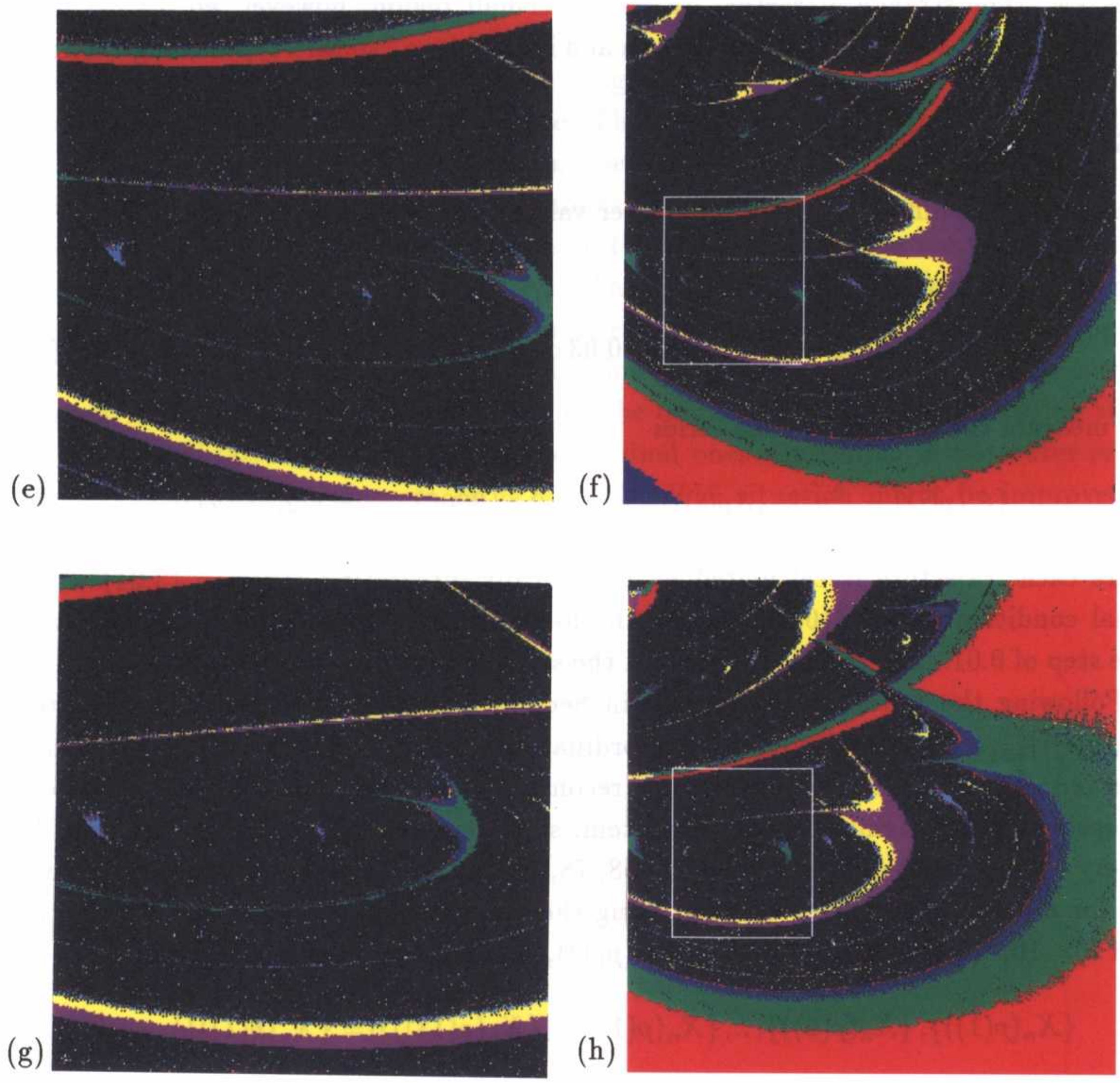


Figure 3.1: (e) (γ_1, γ_2)-bifurcation diagram of the nonlinear predictors (3.20) reconstructed in subsection 3.3.2 with $(\gamma_1, \gamma_2) \in [-0.098, 0.106] \times [-0.044, 0.056]$. (f) $(\gamma_1, -\gamma_2)$-bifurcation diagram of the nonlinear predictors (3.20) reconstructed in subsection 3.3.2 with $(\gamma_1, \gamma_2) \in [-0.169, 0.560] \times [-0.127, 0.171]$. The rectangle region corresponds to the bifurcation diagram of fig. 3.1 (e). (g) (γ_1, γ_2)-bifurcation diagram of the nonlinear predictors (3.20) reconstructed in subsection 3.3.3 with $(\gamma_1, \gamma_2) \in [-0.072, 0.106] \times [-0.042, 0.045]$. (h) $(\gamma_1, -\gamma_2)$-bifurcation diagram of the nonlinear predictors (3.20) reconstructed in subsection 3.3.3 with $(\gamma_1, \gamma_2) \in [-0.134, 0.501] \times [-0.114, 0.145]$. The rectangle region corresponds to the bifurcation diagram of fig. 3.1 (e).

The bifurcation structures shown in figs. 3.1 (a) and (b) have been deeply studied by Gaspard-Kapral-Nicolis [70]. On the (p_1, p_2)-parameter space, it is clearly seen that "fishhook"-like periodic windows form a spiral structure. In the upper-half region of fig. 3.1 (b), eq. (3.24) has a homoclinic orbit which passes through the origin and generates a "screw"-type strange attractor. In the lower-half region, however, eq. (3.24) has no homoclinic orbit with respect to the origin and it exhibits a "spiral"-type strange attractor.

3.3.1 Noise-free experiment

Consider twelve different sets of parameter values (see fig. 3.2 (a)):

$$\begin{aligned} p(i) &= (p_1(i), p_2(i)) \\ &= (\ 0.7\sin(2\pi\frac{(i-1)}{12}) + 5.0,\ 0.03\cos(2\pi\frac{(i-1)}{12}) + 0.33\ )\ (i = 1,2,..,12)(3.28) \end{aligned}$$

and measure the associated time series

$$\{\xi_n(p(i)) = {}^2\eta_{\kappa+n\Delta t}(p(i))) : n = 1,2,..,3000\}\ \ (i = 1,2,..,12) \qquad (3.29)$$

where each trajectory is calculated by numerically integrating equation (3.24) with an initial condition $\eta_0 = {}^T(0.05, 0, 0)$ by the fourth-order Runge-Kutta algorithm with a time step of 0.01. The transient time and the sampling rate are set to $(\kappa, \Delta t) = (2.0, 0.2)$.

Following the procedures described in Section 3.2, we first reconstruct trajectories $\{X_n(p(i))\}_{i=1,2,..,12}$ in a filtered delay-coordinate space with $(d, W, \tau) = (3, 8, 4)$ from the observed chaotic time series, Here, the reconstruction dimension d is set equal to the dimension of the original dynamical system, since one can estimate the dimension by a variety of time series analyses, e.g., [28, 58, 78, 105, 112]. Of course, our procedure works well for d larger than three. Second, using the nonlinear predictors defined by eq. (3.8) with $h = 10$, we seek the parameters $\{\Omega(p(1)), \Omega(p(2)), ..\}$ corresponding to

$$\{X_n(p(1))\}, \{X_n(p(2))\}, .., \{X_n(p(12))\}, \{X_n(p(13))\}(= \{X_n(p(1))\}), ... \qquad (3.30)$$

by minimizing the cost function defined by eq. (3.12) with $K = 4$. Finally, we extract effective parameters $\{\gamma_1, \gamma_2, ..\}$ of Ω by the principal component analysis applied to subsequence

$$\begin{aligned} &\{\Omega(p(N_I)),\ \Omega(p(N_I+1)), ..,\ \Omega(p(N_I+N_J-1))\} \qquad (3.31) \\ &\text{with } (N_I, N_J) = (1440, 24). \end{aligned}$$

The solid line of fig. 3.2 (b) shows the normalized eigenvalues of the covariance matrix (3.16) $\{\Gamma_k = 100 \times \lambda_k / \sum_{j=1}^{70} \lambda_j[\%] : k = 1,2,..,10\}$ and the broken line shows their accumulated sums $\{\sum_{j=1}^{k} \Gamma_j[\%] : k = 1,2,..,10\}$. Since the fig. 3.2 (b) shows that $\sum_{j=1}^{2} \Gamma_j > 90[\%]$, it is clearly seen that the principal component parameters of eq. (3.31) are $\Gamma_2 = (\gamma_1, \gamma_2)$. The dimension of the bifurcation parameter p is therefore correctly estimated as $m = 2$. Fig. 3.2 (c) shows the bifurcation locus in the (γ_1, γ_2)-space. Compared to the bifurcation path of fig. 3.2 (a), the original configuration of the bifurcation path is preserved in the bifurcation locus without any large distortion. Correspondence between the principal parameters and the original ones can be roughly illustrated as $p_1 \leftrightarrow \gamma_1$ and $p_2 \leftrightarrow \gamma_2$.

Fig. 3.1 (c) and (d) shows bifurcation diagrams of the reconstructed family of dynamical systems (3.20) in the (γ_1, γ_2)-space. The local bifurcation diagram of fig. 3.1 (c) reproduces qualitatively similar bifurcation phenomena as the original. The bifurcation structures with continually connected fishhooks are discernible.

Fig. 3.2 (d) shows the bifurcation locus in the (γ_1, γ_3)-space. We see that the amplitude of γ_3 eventually increases for large $|\gamma_1|$. This indicates that the image of the original parameter space p in the Γ-space is not exactly confined in the (γ_1, γ_2)-space. The approximation error due to describing the distorted image by a two-dimensional linear surface may grow in the outer region. This might be the original cause of the bifurcation structures being qualitatively different from the original ones discernible in the outer region of the global bifurcation diagram of fig. 3.1 (d). The situation might be improved by approximating the nonlinear manifold by nonlinear transformations.

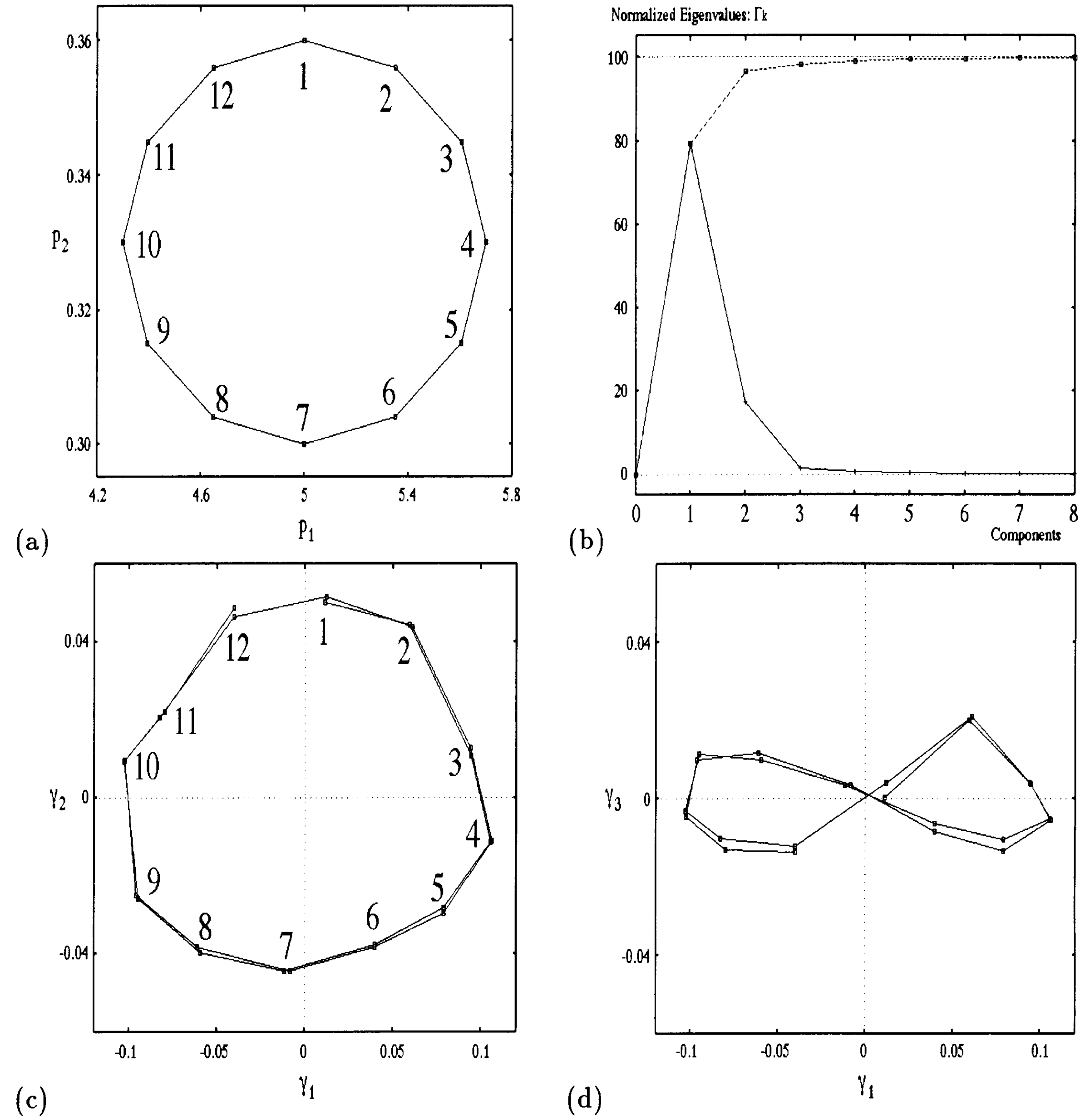


Figure 3.2: (a) Bifurcation path in the (p_1, p_2)-space. (b) Normalized eigenvalues of the principal components $\{\Gamma_k = 100 \times \lambda_k / \sum_{j=1}^{70} \lambda_j[\%] : k = 1, 2, .., 10\}$ (solid line) and their accumulated sums $\{\sum_{j=1}^{k} \Gamma_j[\%] : k = 1, 2, .., 10\}$ (broken line). (c) Bifurcation locus in the (γ_1, γ_2)-space. (d) Bifurcation locus in the (γ_1, γ_3)-space.

3.3.2 Noisy experiment

In order to test the robustness of the algorithm against observational noise, let us perform the same experiment on chaotic time series which include observational noise. The noisy chaotic time series are given by

$$\{\xi_n(p(i)) = \nu_n +^2 \eta_{\kappa+n\Delta t}(p(i)) : n = 1, 2, .., 3000\} \ \ (i = 1, 2, .., 12) \tag{3.32}$$

where the recording conditions are the same as for eq. (3.29) except that the observation includes a Gaussian noise $\{\nu_n\}$ whose mean and standard deviation are set to 0 and 0.1. The order of the bifurcation path of eq. (3.23) is also randomized as (see also fig. 3.3 (a)):

$$\begin{gathered} p(1) \rightarrow p(11) \rightarrow p(4) \rightarrow p(5) \rightarrow p(2) \rightarrow p(3) \rightarrow p(12) \rightarrow \\ p(9) \rightarrow p(10) \rightarrow p(7) \rightarrow p(6) \rightarrow p(8) \rightarrow p(1) \cdots \end{gathered} \tag{3.33}$$

Figs. 3.1 (e) and (f) and figs. 3.3 (a), (b), and (c) show the experimental results obtained by the same algorithm as subsection 3.3.1. Since the results are comparable with the ones obtained by the noise-free experiment, the proposed algorithm seems to be robust against the observational noise. Also, performance of the algorithm is not strongly affected by the choice of the bifurcation path of eq. (3.34).

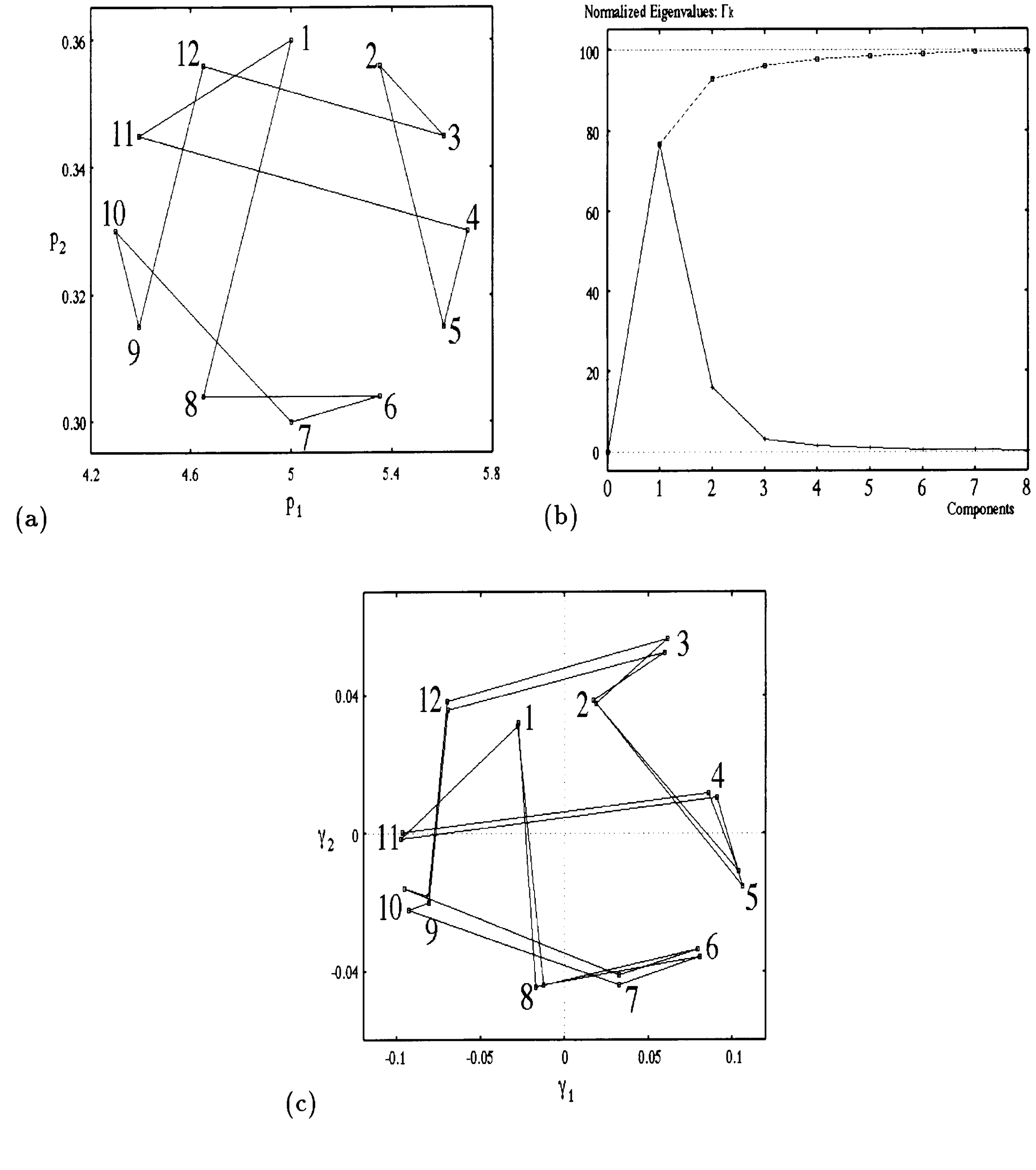


Figure 3.3: (a) Randomized bifurcation path (3.34). (b) Normalized eigenvalues of the principal components $\{\Gamma_k = 100 \times \lambda_k / \sum_{j=1}^{70} \lambda_j [\%] : k = 1, 2, .., 10\}$ (solid line) and their accumulated sums $\{\sum_{j=1}^{k} \Gamma_j [\%] : k = 1, 2, .., 10\}$ (broken line). (c) Bifurcation locus in the (γ_1, γ_2)-space.

3.3.3 Minimum number of chaotic time series required for reconstructing a parametrized family

In the previous two subsections 3.3.1 and 3.3.2, efficiency of our algorithm has been demonstrated under the condition that chaotic time series associated with twelve sets of different bifurcation parameter values $p(i)$ $(i = 1, 2, .., 12)$ are observed. Then, the next question would be

How many chaotic time series are necessary to reconstruct qualitatively similar parametrized family of chaotic dynamics?

Since we deal with two-parameter family of Rössler equations, the minimum condition for reconstructing the parameterized family is $\dim\{p(i)\} = 2$. Hence, the minimum number of chaotic time series, in this case, is three, where the chaotic time series are generated from three sets of parameter values $p(i)$ $(i = 1, 2, 3)$ with triangle distribution.

In order to test this minimum condition, let us perform the same experiment with subsections 3.3.1 and 3.3.2 using chaotic time series associated with three sets of parameter values (see fig. 3.4 (a)):

$$\begin{aligned} p(i) &= (p_1(i), p_2(i)) \\ &= (\ 0.7 \sin(2\pi \frac{(i-1)}{3}) + 5.0,\ 0.03 \cos(2\pi \frac{(i-1)}{3}) + 0.33\ )\ (i = 1, 2, 3). \end{aligned} \tag{3.34}$$

In this experiment, no observational noise is included in the chaotic time series.

Figs. 3.1 (g) and (h) and figs. 3.4 (a), (b), and (c) show the experimental results obtained by the same algorithm with the subsection 3.3.1. Since the results are comparable with the ones obtained by using twelve sets of chaotic time series, our algorithm seems to be efficient also when only three sets of chaotic time series are given. The results, therefore, imply that the minimum number of chaotic time series required for reconstructing a 2-parameter family of Rössler equations is three.

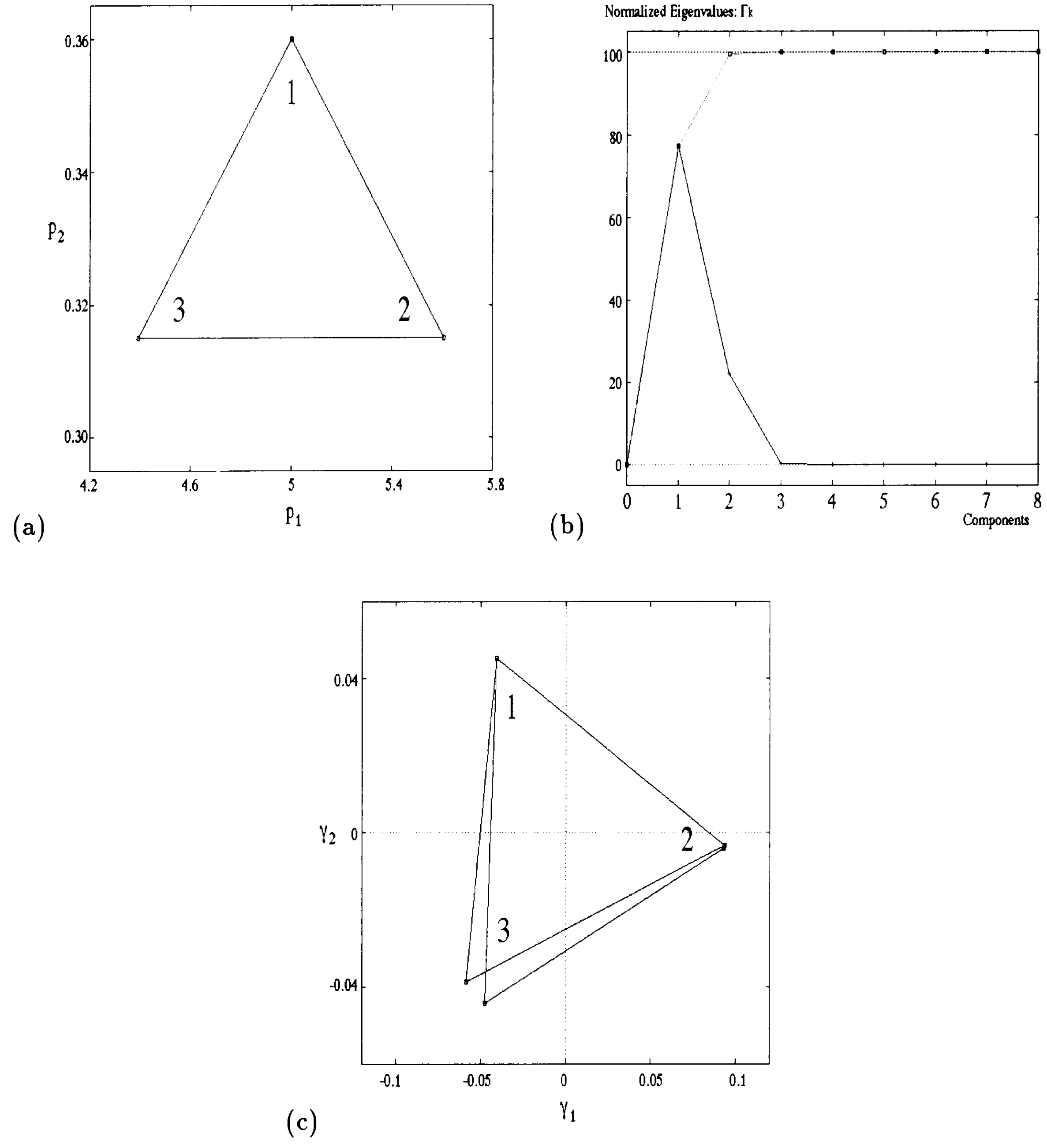


Figure 3.4: (a) Bifurcation path in the (p_1,p_2)-space. (b) Normalized eigenvalues of the principal components $\{\Gamma_k = 100 \times \lambda_k/\sum_{j=1}^{70}\lambda_j[\%] : k = 1,2,..,10\}$ (solid line) and their accumulated sums $\{\sum_{j=1}^{k}\Gamma_j[\%] : k = 1,2,..,10\}$ (broken line). (c) Bifurcation locus in the (γ_1,γ_2)-space.

3.4 Recognizing chaotic time series

In this Section, we conduct an experimental study on "chaotic time series recognition" based on the qualitatively similar parametrized family of nonlinear predictors constructed in the previous Section. Here, we estimate the original bifurcation parameter values $p*$ in terms of its image in the principal component parameter space $\Gamma_2(p*)$ only from the associated chaotic time series

$$\{\xi_n(p*) =^2 \eta_{\kappa+n\Delta t}(p*) : n = 1, 2, .., 3000\}, \tag{3.35}$$

where the above data is recorded in the same condition as eq. (3.29).

The corresponding bifurcation parameters $\Gamma_2(p*)$ are sought by minimizing the cost function

$$U(\Gamma_2) = \sum_{n=(d-1)\tau+1}^{N-W-K} \sum_{k=1}^{K} \frac{1}{2} \mid X_{n+k}(p*) - \phi^{k\Delta}(\Gamma_2, X_n(p*)) \mid^2 \tag{3.36}$$

where $\phi^t : R^2 \times R^d \to R^d$ stands for a solution of eq. (3.20) at the parameter values Γ_2 with an initial condition $\phi^0(\Gamma_2, X) = X$.

For 48 sets of bifurcation parameter values (see fig. 3.5 (a)):

$$\begin{aligned} p(i) &= (p_1(i), p_2(i)) \\ &= (\ R_1 \cos(2\pi\frac{(i-1)}{12}) + 5.0,\ R_2 \sin(2\pi\frac{(i-1)}{12}) + 0.33\ )\ \ (i = 1, 2, .., 12) \\ \text{with } (R_1, R_2) &= (0.35, 0.015), (0.7, 0.03), (1.4, 0.06), (2.1, 0.09), \qquad (3.37) \end{aligned}$$

the corresponding parameters $\{\Gamma_2(p(i))\}$ in the Γ_2-space are estimated. Fig. 3.5 (b) shows the result for noiseless data; fig. 3.5 (c) shows the result for noisy data.

From the similar configurations discernible in the locations of the corresponding bifurcation parameters in the p-space and the Γ_2-space, we see that parameters qualitatively similar to the original p can be estimated in the Γ_2-space.

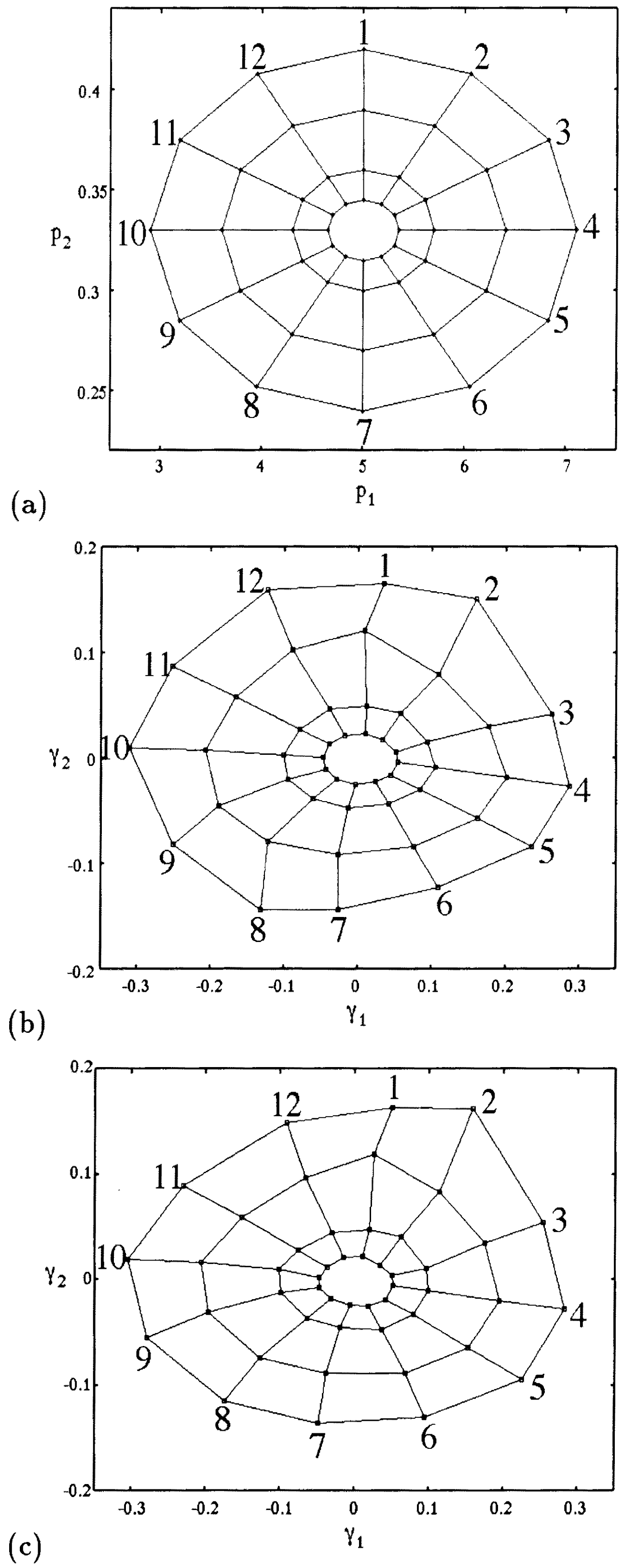


Figure 3.5: Experiment on chaotic time series recognition. (a) Selected parameters in the (p_1, p_2)-space. (b) Estimated parameters in the (γ_1, γ_2)-space by noiseless experiment. (c) Estimated parameters in the (γ_1, γ_2)-space by noisy experiment.

3.5 Conclusions and discussions

In this Chapter, an algorithm is proposed for constructing a parametrized family of nonlinear predictors "qualitatively similar" to a family of continuous-time dynamical systems from only several sets of chaotic time series. Several numerical experiments using the Rössler family show efficiency of the algorithm. On the basis of the family qualitatively similar to the Rössler equations, we have shown that sets of chaotic time series associated with unknown bifurcation parameter values can be systematically characterized in terms of the qualitative bifurcation parameters of the nonlinear predictors. Our experiments therefore demonstrate the applicability of the method for "chaotic time series recognition."

By setting various experimental conditions, we have shown that the algorithm is not only robust against observational noise but it is also effective to reconstruct a 2-parameter family of Rössler equations when only three sets of chaotic time series are provided.

Since we have confirmed efficiency of the algorithm to equational models of chaotic dynamics, our future work will be devoted for applying the algorithm to real world dynamical systems such as electronic circuits [135], chaotic chemical sensor [169], and chaotic neural oscillators [9, 134].

Chapter 4

Detecting switch dynamics in chaotic time series

abstract: An algorithm is presented for detecting switch dynamics in chaotic time series. By the "switch dynamics," we mean that the chaotic time series is measured from a dynamical system whose bifurcation parameters are occasionally switched among a set of slightly different parameter values. First, the switched chaotic time series is divided into windows of short-term time series. From the set of windowed time series, "qualitatively similar" parametrized family of neural networks is constructed using the algorithm presented in Chapter 3. By characterizing the windows of short-term chaotic time series in terms of the "qualitative" parameters of neural networks, switch dynamics of their associated bifurcation parameters are detected. For the Lorenz equations, the Rössler equations, and the Mackey-Glass equations, efficiency of the algorithm is demonstrated.

4.1 Introduction

In the studies of chaotic time series analyses [1, 32, 41, 51, 54, 57, 78, 101, 123, 171, 182, 212], it has been supposed that the time series is stationary and the bifurcation parameters of the underlying chaotic dynamical system are not changed. Such chaotic time series can be characterized by the statistical property of the underlying chaotic dynamics such as fractal dimension [78], Lyapunov exponents [51, 171, 212], Kolomogorov-Sinai entropy [57], and nonlinear predictability [32, 41, 54, 123, 182]. In practice, however, bifurcation parameters of the time series measured from real-world systems can be occasionally changed. For example, in the flow dynamics, the system variables are sometimes composed of the fast-dynamic components and the slow-dynamic ones [40]. The dominant fast-dynamic patterns are occasionally changed by the slow-dynamic variables, which can be considered as the occasionally changed bifurcation parameters.

The occasional change in the bifurcation parameters can also have a functionality for transmitting binary code information. For example, human speech with binary word information is transmitted by the sequential changes in the oscillatory states of the vocal system. Another example is the chaotic secret communication systems [42, 44, 161]. This communication system is composed of transmitter subsystem and receiver subsystem. With a successive change between two bifurcation parameter values, p_+ and p_-, the transmitter subsystem exhibits switch dynamics among the two chaotic attractors associated with p_+ and p_-. By the Pecora-Carroll synchronization [31, 159], the switch dynamics of the transmitter is sent to the receiver and the binary information encoded as a sequence of the switched bifurcation parameters can be decoded in the synchronized receiver subsystem.

For the analyses of chaotic time series with such switched bifurcation parameters, conventional techniques for analyzing chaotic time series can not be directly applied. This is because the conventional techniques mainly estimate statistical properties of the underlying chaotic dynamics from long enough time series data with fixed bifurcation parameters. Towards the analyses of chaotic time series with switched bifurcation parameters, various new numerical algorithms have been recently developed [33, 104, 111, 133, 177, 178, 211]. The basic numerical procedure for analyzing switch dynamics in chaotic time series is to first divide a time series into windows of short-term time series data. Then dynamical closeness between the windows of the data are measured by computing the difference in statistical quantity between the windowed data such as invariant measure [104, 111, 211], cross-correlation integral [133, 178], recurrence plot [33, 133], and cross prediction error [177]. Based on the statistical test which detects a significant change in the statistical quantity of the chaotic time series, the stationarity of the data can be examined.

Although these algorithms have been successfully applied to various chaotic time series with switched bifurcation parameters, there might be some limitations due to the following problems:

1) If the switch interval of the bifurcation parameters is short, reliable estimation of the statistical quantities from such short-term data can not be always expected.

2) If the switched bifurcation parameter values are closely located with others and if the time series data is contaminated with observational noise, qualitative dynamics as well as the statistical properties of the associated chaotic attractors might be similar to each other. Hence, detection of a slight change in the statistical property of the switched chaotic time series might be quite difficult.

Our approach to the problem is rather different from the techniques of [33, 104, 111, 133, 177, 178, 211]. Since the qualitative change in the chaotic time series is induced by the switch in the bifurcation parameter values, it is natural and more efficient to detect the switch dynamics by estimating the underlying switched bifurcation parameters.

The problem for estimating the underlying bifurcation parameters from chaotic time series has been studied in Chapter 3. Since it is supposed that there is no information about the functional form of the parameterized family of chaotic dynamics, estimation of the exact bifurcation parameter values only from time series is impossible. Instead, "qualitatively similar" bifurcation parameter values can be estimated by a simple algorithm using a parametrized family of nonlinear predictors. The "qualitatively similar" bifurcation parameters mean that the parameters give rise to a family of nonlinear predictors which exhibits qualitatively similar bifurcation phenomena as the original. The algorithm has been successfully applied to the Rössler family.

Based on the estimation technique of the underlying bifurcation parameters, this Chapter presents an algorithm for detecting switch dynamics in chaotic time series. Using three typical chaotic dynamical systems, the Lorenz equations, the Rössler equations, and the Mackey-Glass equations, efficiency of the algorithm is demonstrated. In the experiments, switched chaotic time series contaminated with observational noise is considered.

4.2 Algorithm for detecting switch dynamics in chaotic time series

4.2.1 Problem formulation

Consider a continuous-time chaotic dynamical system:

$$\frac{d\eta_t}{dt} = f(p(s(t)), \eta_t), \quad p \in R^m, \quad \eta_t \in R^D, \tag{4.1}$$

and its observation:

$$\{\xi_t = g(\eta_t) \mid 0 \leq t \leq C\}, \tag{4.2}$$

where the bifurcation parameter $p(s(t))$ is occasionally changed among I-different sets of parameter values $\{p(i)\}_{i=0,1,..,I-1}$ by the switch signal $s(t)$,

$$s(t) = \begin{cases} 0, & \text{for } t \in V_0 \quad (\subset[0, C]) \\ 1, & \text{for } t \in V_1 \\ \vdots, & \\ I-1, & \text{for } t \in V_{I-1} \end{cases} \tag{4.3}$$

where $\cup_{i=0}^{I-1} V_i = [0, C]$ and $V_i \cap V_j = \emptyset$ for $i \neq j$.

Here we assume the followings:

(i) The functional form of the parametrized family of vector fields $f : R^m \times R^D \to R^D$ is not known; f is assumed to be smooth.

(ii) The functional form of $g : R^D \to R^1$ is not known; g is assumed to be smooth.

(iii) $D = \dim \eta_t$ is not known.

(vi) $m = \dim p$ and the sets of parameter values $\{p(i)\}_{i=0,..,I-1}$ are not known; $\{p(i)\}_{i=0,..,I-1}$ are closely located with each other.

Under the conditions (i)-(vi), we consider an algorithm for detecting switch dynamics of the bifurcation parameters p in the chaotic time series $\{\xi_t \mid 0 \le t \le C\}$. The algorithm is composed of four steps. First, observational noise in the measured time series is smoothed out by an averaging filter, and high-dimensional chaotic trajectory is reconstructed using the delay-coordinate method. Second, the chaotic trajectory is divided into windows of short-term trajectories, and nonlinear predictors which model the windowed chaotic trajectories are constructed within a same parametrized family. Third, effective bifurcation parameters are extracted from the many parameters of the nonlinear predictors by principal component analysis. Fourth, windows of shorter-term chaotic trajectories are characterized by the principal bifurcation parameters of the nonlinear predictors and the switch dynamics in the principal bifurcation parameters are detected by the Linde-Buzo-Gray (LBG) clustering algorithm.

4.2.2 Averaged filtering and delay-coordinate embedding

In laboratory experiments, time series are usually sampled digitally and also contaminated with observational noise. Hence, let us rewrite the eq. (4.2) by

$$\{\xi_n = g(\eta_{n\Delta t}) + \nu_n \mid n = 1, 2, \cdots, N_{data}\} \tag{4.4}$$

where Δt is the sampling rate, N_{data} is the number of the data, and ν_n is a Gaussian noise.

In order to smooth out the observational noise, an averaging filter is applied to the time series as

$$\{\hat{\xi}_n = \frac{1}{W+1} \sum_{k=n}^{n+W} \xi_k \mid n = 1, 2, \cdots, N_{data} - W\}, \tag{4.5}$$

where W is the window length of the moving average.

From the filtered time series $\{\hat{\xi}_n \mid n = 1, 2, \cdots, N - W\}$, a d-dimensional trajectory $\{X_n \mid n = 1 + (d-1)\tau, \cdots, N_{data} - W\}$ is reconstructed by using a delay-coordinate [186, 176]:

$$\begin{aligned} X_n &= {}^{T}({}^{1}x_n, {}^{2}x_n, \cdots, {}^{d}x_n) \\ &= {}^{T}(\hat{\xi}_n, \hat{\xi}_{n-\tau}, \cdots, \hat{\xi}_{n-(d-1)\tau}), \end{aligned} \tag{4.6}$$

where T denotes transposition and τ denotes time lag. The Filtered Delay Embedding Prevalence Theorem [176] guarantees that the reconstructed trajectory $\{X_n\}$ is qualitatively the same as the original $\{\eta_t\}$.

4.2.3 Nonlinear predictors

In order to detect qualitative change in the reconstructed trajectory $\{X_n\}$, we divide the trajectory $\{X_n\}$ into J-windows of short-term trajectories with T-interval:

$$\{X_n(i) = X_n \mid n = 1 + (d-1)\tau + (i-1)\,T, \cdots, (d-1)\tau + i\,T\}_{i=1,2,..,J}. \tag{4.7}$$

For each windowed trajectory, $\{X_n(i)\}_{i=1,2,..,J}$, we construct a nonlinear predictor $F : R^L \times R^d \to R^d$ which model the trajectory dynamics as

$$X_{n+1}(i) \approx F(\Omega(i), X_n(i)), \tag{4.8}$$

where $\Omega \in R^L$ stands for a set of parameters of nonlinear predictor $F(\cdot,\cdot)$. As a nonlinear predictor, multi-layer perceptron (MLP) [167] is exploited, where the MLP $\tilde{f}$ composed of three-layers (d-units in the input layer, d-units in the output layer, and h-units in the hidden layer) is given by

$$\tilde{f}(\Omega, X) = {}^T(\ \tilde{f}_1(\Omega, X), \tilde{f}_2(\Omega, X), \cdots, \tilde{f}_d(\Omega, X)\ ) \tag{4.9}$$

where

$$\begin{aligned}
\tilde{f}_k(\Omega, X) &= \sum_{j=1}^{h} \omega_{(k-1)h+j}\ \sigma(\sum_{i=1}^{d} \omega_{dh+(j-1)d+i}\ {}^i x + \omega_{2dh+j}\ )\ \ (k = 1, 2, .., d), \\
\sigma(y) &= \frac{2}{1+e^{-y}} - 1, \\
\Omega &= {}^T(\omega_1, \omega_2, \cdots, \omega_L) \ \text{ with } \ L = (2d+1)h.
\end{aligned}$$

Using the MLP, the nonlinear predictor F is constructed as

$$F(\Omega, X) = X + \Delta t \tilde{f}(\Omega, X). \tag{4.10}$$

The parameters $\{\Omega(i)\}_{i=1,2,..,J}$ which correspond to the windowed trajectories $\{X_n(i)\}_{i=1,2,..,J}$ are computed as follows. First, J-windows of trajectories $\{X_n(i)\}_{i=1,2,..,J}$ are periodically ordered as

$$\{X_n(1)\}, \{X_n(2)\}, \cdots, \{X_n(J)\}, \{X_n(J+1)\}(= \{X_n(1)\}), \cdots. \tag{4.11}$$

Second, $\Omega(1)$ is computed by minimizing the cost function:

$$U(\Omega) = \sum_{n=1+(d-1)\tau}^{(d-1)\tau+T-K} \sum_{k=1}^{K} \frac{1}{2} \mid X_{n+k}(1) - F^k(\Omega, X_n(1)) \mid^2 \tag{4.12}$$

via the gradient-descent method with a random initial condition $\Omega(1) \in [0,1]^L$. * Then, $\Omega(i)$ $(i = 2, 3, \cdots)$ is computed by minimizing the cost function (4.12) defined for the i-th

* In our experiment, the cost function (4.12) is minimized by a single iteration of the gradient-descent procedure $\Omega' = \Omega - \alpha \nabla U(\Omega)$, where $\nabla U(\Omega)$ is a gradient vector and α is determined by the line-search method which minimizes $U(\Omega - \alpha \nabla U(\Omega))$. The gradient vector $\nabla U(\Omega)$ is computed by the recurrent back-propagation algorithm of [209].

trajectory $\{X_n(i)\}$ in a similar manner as $\Omega(1)$ except that $\Omega(i-1)$ is selected as the initial condition instead of the random values.

The procedures for computing $\{\Omega(i)\}_{i=1,2,..}$ are repeated until they converge to a periodic sequence as

$$\Omega(N_J), \Omega(N_J+1), \cdots, \Omega(N_J+J), \Omega(N_J+J+1)(=\Omega(N_J)), \cdots \tag{4.13}$$

where N_J is assumed to be sufficiently large.

4.2.4 Extracting principal bifurcation parameters

From the nonlinear prediction parameters $\{\Omega(i)\}$, principal component parameters are extracted by the Karuhnen-Loéve (KL) transform [14, 21, 28, 140].

First, we consider the subsequence of the parameters $\{\Omega(i) \mid i = N_J, N_J+1, \cdots, N_J + N_K - 1\}$ (N_K : element number) and compute Ω_0 and $\{\delta\Omega_i \mid i = 1, 2, .., N_K\}$ as

$$\begin{aligned} \Omega_0 &= \frac{1}{N_K}\sum_{i=1}^{N_K} \Omega(N_J+i-1), & (4.14) \\ \delta\Omega_i &= \Omega(N_J+i-1) - \Omega_0. & (4.15) \end{aligned}$$

Second, the multivariate distribution of $\{\delta\Omega_i \mid i = 1, 2, .., N_K\}$ is computed in terms of the covariance matrix:

$$\Omega_{L\times L} = \frac{1}{N_K}\sum_{i=1}^{N_K} \delta\Omega_i \ {}^T\delta\Omega_i. \tag{4.16}$$

Since $\Omega_{L\times L}$ has non-negative eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_L\}$, they are arranged in descending order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0. \tag{4.17}$$

Applying KL-transformation to $\delta\Omega$, the principal parameters are given by

$$\Gamma = (\gamma_1, \gamma_2, \cdots, \gamma_L) = {}^T[u_1 \mid u_2 \mid \cdots \mid u_L]^{-1}\, \delta\Omega \tag{4.18}$$

where $\{u_1, u_2, \cdots, u_L\}$ are the eigenvectors corresponding to $\{\lambda_1, \lambda_2, \cdots, \lambda_L\}$.

Since the transformation (4.18) diagonalizes the covariance matrix (4.16) in Γ-space, the diagonal elements $\{\lambda_1, \lambda_2, \cdots, \lambda_L\}$ represent the significance of their corresponding principal parameters $\{\gamma_1, \gamma_2, \cdots, \gamma_L\}$. By computing the normalized eigenvalues

$$\Lambda_i = \frac{\lambda_i}{\sum_{j=1}^{L}\lambda_j} \quad (i = 1, 2, .., L), \tag{4.19}$$

and the cumulative significance ratio from the first to the k-th principal components

$$\Theta_k = \sum_{i=1}^{k}\Lambda_i \quad (k = 1, 2, .., L), \tag{4.20}$$

the number of significant parameters M (empirically set as $\Theta_M > 0.98$) is determined.

With respect to the significant parameters Γ_M, the M-parameter family of nonlinear predictors is constructed as

$$X_{n+1} = F(\Omega(\Gamma_M), X_n) \tag{4.21}$$

where

$$\Omega(\Gamma_M) = {}^{T}[u_1 \mid u_2 \mid \cdots \mid u_L]^{-1}[\Gamma_M \mid 0] + \Omega_0 \tag{4.22}$$

where 0 denotes $(L - M)$-dimensional 0 column vector.

In Chapter 3, it has been shown that the principal parameter family of nonlinear predictors (4.21) exhibits qualitatively similar bifurcation phenomena as the original dynamics (4.1), where the original bifurcation parameters p are mapped to the principal parameters Γ_m via a homeomorphism $\psi : R^m \rightarrow R^m$. This implies that the switch dynamics in the original bifurcation parameters p can be detected in the principal parameter space of the nonlinear predictors (4.21).

4.2.5 Detection of switch points

In order to detect switch dynamics in the chaotic trajectory $\{X_n\}$, we characterize the temporal dynamics of the chaotic trajectory in terms of the principal bifurcation parameters Γ_M. Again, we divide the trajectory $\{X_n\}$ into Q-windows of shorter-term trajectories with S-interval:

$$\{X_n(i) = X_n \mid n = 1 + (d-1)\tau + (i-1)\ S, \cdots, (d-1)\tau + i\ S\}_{i=1,2,..,Q}. \tag{4.23}$$

Then, each window of trajectory $\{X_n(i)\}$ is characterized by the principal bifurcation parameters $\Gamma_M(i)$ which approximate the trajectory dynamics as

$$X_{n+1}(i) \approx F(\Omega(\Gamma_M(i)), X_n(i)). \tag{4.24}$$

The principal parameters $\Gamma_M(i)$ can be computed by minimizing the cost function:

$$U(\Gamma_M) = \sum_{n=1+(d-1)\tau+(i-1)S}^{(d-1)\tau+iS-K} \sum_{k=1}^{K} \frac{1}{2} \mid X_{n+k}(i) - F^k(\Omega(\Gamma_M), X_n(i)) \mid^2 \tag{4.25}$$

via the quasi-Newton method [132] with random initial condition $\Gamma_M \in [0,1]^M$.

Finally, in order to determine the number of switch parameters I and to classify the sets of principal parameters $\{\Gamma_M(i) \mid i = 1, 2, .., Q\}$ into the corresponding switched parameters $\{p(i) \mid i = 0, 1, .., I - 1\}$, the LBG-clustering algorithm [128] is applied. The algorithm analyzes the distribution of the principal parameters $\{\Gamma_M(i) \mid i = 1, 2, .., Q\}$

and classify them into q-nonoverlapping subgroups $\{\Gamma_M(i) \mid i \in R_j\}_{j=0,1,..,q-1}$ ($R_j \neq \emptyset$, $\cup_{j=0}^{q-1} R_j = \{1,2,..,Q\}$, $R_i \cap R_j = \emptyset$ for $i \neq j$), which minimizes the distortion function:

$$\begin{aligned} D_q(R_1, R_2, \cdots, R_q) &= \frac{1}{Q\,\Xi} \sum_{j=0}^{q-1} \sum_{i \in R_j} \mid \Gamma_M(i) - \hat{\Gamma}(j) \mid^2, \\ \Xi &= \max_{(i,j)} \mid \Gamma_M(i) - \Gamma_M(j) \mid^2, \\ \hat{\Gamma}(j) &= \frac{1}{N_j} \sum_{i \in R_j} \Gamma_M(i) \quad (N_j\text{: number of elements in } R_j), \end{aligned} \tag{4.26}$$

where Ξ is a normalization constant and $\hat{\Gamma}(j)$ is the j-th centroid.

Using the least number of clusters q_{opt} which provides sufficiently small distortion function D_q (empirically set as $D_{q_{opt}} < 0.01$), we can determine the number of the switch parameters and classify the principal parameters $\{\Gamma_M(i) \mid i = 1,2,..,Q\}$ into the corresponding centroids $\{\hat{\Gamma}(i) \mid i = 0,1,..,q_{opt} - 1\}$. If $q_{opt} = I$ and the switch centroids $\{\hat{\Gamma}(i) \mid i = 0,..,q_{opt} - 1\}$ have one-to-one correspondence with the original switch points $\{p(i) \mid i = 0,1,..,I-1\}$, the switch dynamics of $s(t)$ can be correctly detected. For our convenience, the detected switch centroids $\{\hat{\Gamma}(0), \hat{\Gamma}(1), \cdots\}$ are denoted by binary signal $s' = 0,1,\cdots$, respectively. Of course, there is an indeterminacy in the permutation of the switch signal $s'(t)$ and exactly the same switch signal as the original $s(t)$ can not be usually recovered. The switch points are finally determined as the time when the switch signal $s'(t)$ changes into another signal as $s'(t) \neq s'(t + \Delta t)$.

4.3 Numerical experiments

In this Section, we test our algorithm against three chaotic dynamical systems: the Lorenz equations [130], the Rössler equations [165], and the Mackey-Glass equations [137]. It is shown that the algorithm detects switch dynamics among two or three sets of bifurcation parameter values.

4.3.1 Lorenz equation

As a first example, we consider the Lorenz equations [130]:

$$\begin{aligned} \frac{d\,{}^1\eta_t}{dt} &= \sigma\,({}^2\eta_t - {}^1\eta_t), \\ \frac{d\,{}^2\eta_t}{dt} &= r\,{}^1\eta_t - {}^2\eta_t - {}^1\eta_t\,{}^3\eta_t, \\ \frac{d\,{}^3\eta_t}{dt} &= {}^1\eta_t\,{}^2\eta_t - b(s(t))\,{}^3\eta_t. \end{aligned} \tag{4.27}$$

In this experiment, parameter values for σ and r are fixed to

$$\sigma = 16,\ r = 45.6,$$

and the bifurcation parameter b makes switches among two values,

$$b(0) = 4.4, \quad b(1) = 4, \tag{4.28}$$

according to the switch signal $s(t)$.

Let us analyze the switch dynamics of the Lorenz equations modulated by the square wave signal $s(t)$ of fig. 4.1 (a). The chaotic time series is then obtained as

$$\{\xi_n = ({}^1\eta_{n\Delta t}/30) + \nu_n \mid n = 1, 2, .., N_{data}\}, \tag{4.29}$$

where the sampling rate, the number of the data, and the Gaussian noise level are set as $\Delta t = 0.02$, $N_{data} = 7200$, and $\nu_n \in N(0, 0.02)$. The Lorenz equation is numerically integrated by the fourth-order Runge-Kutta algorithm with a time step of 0.001.

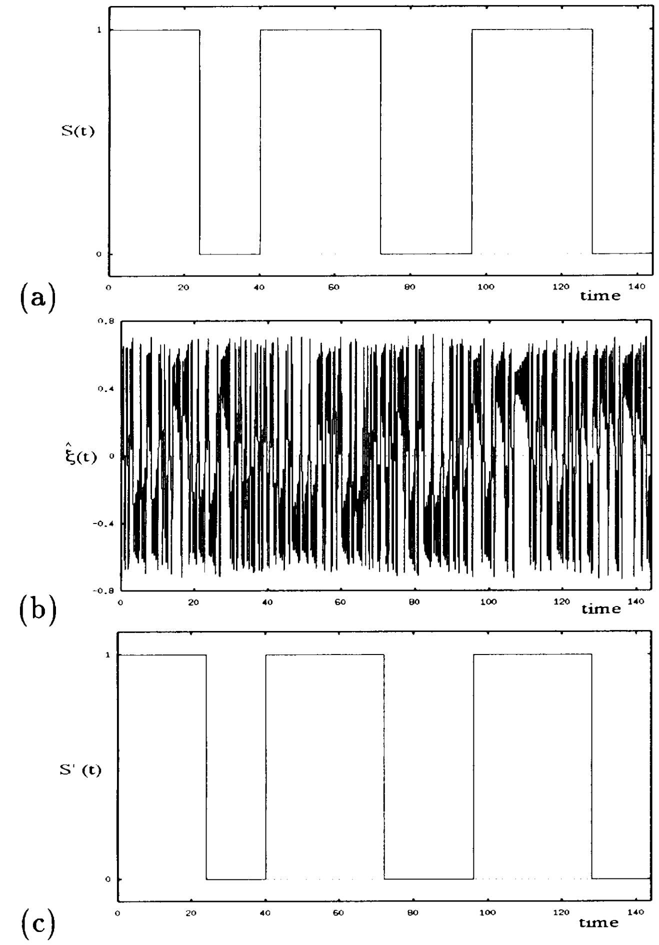


Figure 4.1: (a) Switch signal $s(t)$ of the bifurcation parameter b in the Lorenz equation (4.27). (b) Average-filtered time series $\{\hat{\xi}_n\}$ (4.5) recorded from the switched Lorenz equation. (c) Switch signal $s'(t)$ predicted by the present algorithm.

Fig. 4.1 (b) shows the average-filtered time series $\{\hat{\xi}_n\}$ with the averaging window length of $W = 7$. While the observational noise has been smoothed out by the average filter, it is difficult to recognize qualitative change in the switched chaotic time series.

When the switch dynamics takes place among chaotic attractors with distinctively different geometric structures, it is reported in [161] that the switch dynamics are discernible in the maximum recurrent plots of the time series. The maximum plots display the geometric difference of the switching attractors and indicate the switch dynamics which falls in either branch of the distinctive attractors. Figs. 4.2 show the maximum plots obtained from the average-filtered time series $\{\hat{\xi}_n\}$ of fig. 4.1 (b). Whereas the two chaotic attractors with $b = 4.4$ and $b = 4$ may have rather different geometric structures, the observational noise thickens their sheet geometries and mixes the domains of the switching attractors. Hence, it is hard to distinguish the attractors and to detect the switching points in the presence of noise. As the number of the switching attractors increases further, systematic detection of switching attractors by simple maximum plots may become much more difficult.

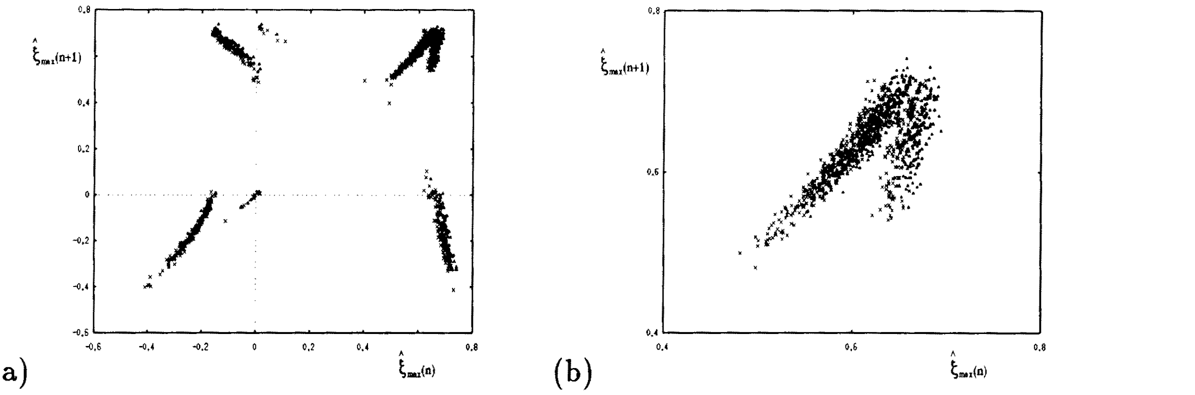


Figure 4.2: (a) Maximum recurrent plots $(\hat{\xi}_{max}(n), \hat{\xi}_{max}(n+1))$ of the average-filtered time series of fig. 4.1 (b). The crosses indicate the maximum plots of the Lorenz equation (4.27) with $b = 4.0$ and the triangles indicate the maximum plots of the Lorenz equation with $b = 4.4$. (b) Enlargement of fig. 4.2 (a) with $(\hat{\xi}_{max}(n), \hat{\xi}_{max}(n+1)) \in [0.4, 0.8] \times [0.4, 0.8]$.

Let us test our algorithm. First, 3-dimensional trajectory $\{X_n \mid n = 1+(d-1)\tau, .., N-W\}$ $(d = 3, \tau = 4)$ is reconstructed from the filtered time series $\{\hat{\xi}_n\}$. The trajectory $\{X_n\}$ is divided into 6-windows of trajectories $\{X_n(i)\}_{i=1,2,..,6}$ with a time interval of $T = 1200$.

Second, using the nonlinear predictors defined by eq. (4.10) with $h = 10$, we seek the parameters $\{\Omega(1), \Omega(2), ..\}$ corresponding to

$$\{X_n(1)\}, \{X_n(2)\}, , .., \{X_n(6)\}, \{X_n(7)\}, (= \{X_n(1)\}), ... \tag{4.30}$$

by minimizing the cost function defined by eq. (4.12) with $K = 2$.

Third, we extract effective parameters $\{\gamma_1, \gamma_2, ..\}$ of Ω by the principal component analysis applied to subsequence $\{\Omega(9988),\ \Omega(9989), ..,\ \Omega(9999)\}$. Fig. 4.3 (a) shows the cumulative significance ratio Θ_k of the covariance matrix (4.16). Since $\Theta_2 > 0.98$, we set the principal component parameters as $\Gamma_2 = (\gamma_1, \gamma_2)$. Fig. 4.3 (b) shows the locations of the subsequence of the parameters $\{\Omega(9988), .., \Omega(9999)\}$ in the (γ_1, γ_2)-space.

Fourth, the trajectory $\{X_n\}$ is divided again into 18-windows of shorter trajectories $\{X_n(i)\}_{i=1,2,..,18}$ with a time interval of $S = 400$. In fig. 4.3 (d), the trajectories $\{X_n(i)\}_{i=1,2,..,18}$ are characterized by the principal parameter values $\{\Gamma_2(i)\}_{i=1,2,..,18}$ which minimize the cost functions (4.25). The switch dynamics among the 2-clusters of distinctive points in prinçipal parameter space is clearly recognized. It is indeed shown in fig. 4.3 (c) that the LBG-clustering is optimized by the cluster number of $q_{opt} = 2$ which gives a sufficiently small distortion function $D_2 < 0.01$. Fig. 4.1 (c) shows the sequence of the LBG-clustering signal $s'(t)$, which predicts the original signal $s(t)$ with good accuracy. Hence, systematic detection of the number of switch dynamics as well as their switch points is realized by the present algorithm.

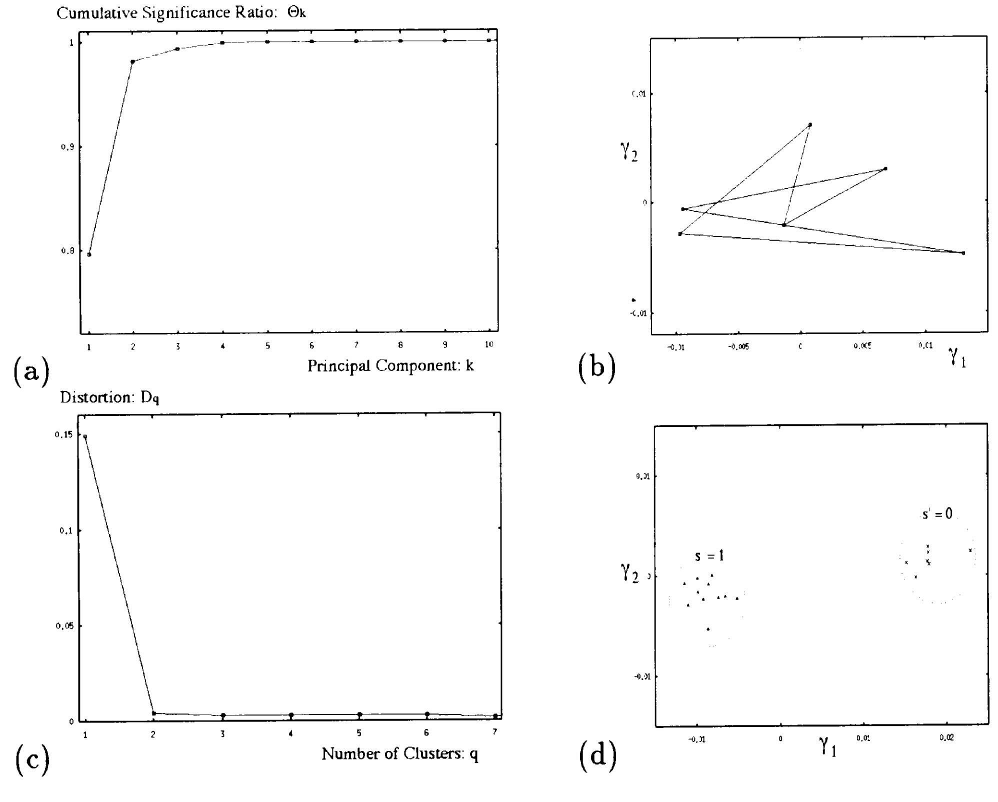


Figure 4.3: (a) Cumulative significance ratio $\{\Theta_k\}$ of the covariance matrix (4.16). (b) Locations of the subsequence $\{\Omega(9988), .., \Omega(9999)\}$ of the nonlinear prediction parameters in the 2-dimensional principal space (γ_1, γ_2). (d) Distortion function D_q (4.27) optimized by the LBG-clustering algorithm with a cluster number q. (d) Locations of the principal parameter values $\{\Gamma_2(i)\}_{i=1,2,..,18}$ corresponding to the windows of short-term trajectories $\{X_n(i)\}_{i=1,2,..,18}$. The principal parameters are classified into 2-groups, where the crosses indicate the points classified into "$s' = 0$" and the triangles indicate the points classified into "$s' = 1$."

It should be noted that, in the present experiment, switch in the bifurcation parameters does not occur within any window of shorter-term chaotic trajectory. If a switch occurs within a window, the principal parameter values Γ_2 corresponding to the window can not be accurately estimated. If such a switch occurs frequently and if the principal parameter values can not be accurately estimated for many windows, identification of the number of the switch dynamics as well as their switch points may become quite difficult. We consider, however, that if the switch occurs only intermittently and if the bifurcation parameters rarely change within a window, reliable estimation of the principal bifurcation parameters is possible for "most" of the windows of chaotic trajectories. Hence, for such intermittent switch signal, the present algorithm may identify the number of switch dynamics with good accuracy.

4.3.2 Rössler equation

As a second example, we consider the Rössler equations [165]:

$$\begin{aligned}\frac{d\,{}^1\eta_t}{dt} &= -\,{}^2\eta_t - {}^3\eta_t, \\ \frac{d\,{}^2\eta_t}{dt} &= {}^1\eta_t + a(s(t))\ {}^2\eta_t, \\ \frac{d\,{}^3\eta_t}{dt} &= b\,{}^1\eta_t - \left(c(s(t)) - {}^1\eta_t\right){}^3\eta_t,\end{aligned} \tag{4.31}$$

where the parameter value b is fixed as $b = 0.3$ and the bifurcation parameters $(a,\ c)$ take three sets of values:

$$(a(0), c(0)) = (0.34, 5.6),\ \ (a(1), c(1)) = (0.36, 5.2),\ \ (a(2), c(2)) = (0.34, 4.8), \tag{4.32}$$

according to the switch signal $s(t)$ of fig. 4.4 (a). The chaotic time series is obtained as

$$\{\xi_n = ({}^2\eta_{n\Delta t}/10) + \nu_n \mid n = 1, 2, .., N_{data}\}, \tag{4.33}$$

where $\Delta t = 0.2$, $N_{data} = 7200$, $\nu_n \in N(0.0, 0.02)$, and the Rössler equation is numerically integrated by the fourth-order Runge-Kutta algorithm with a time step of 0.01.

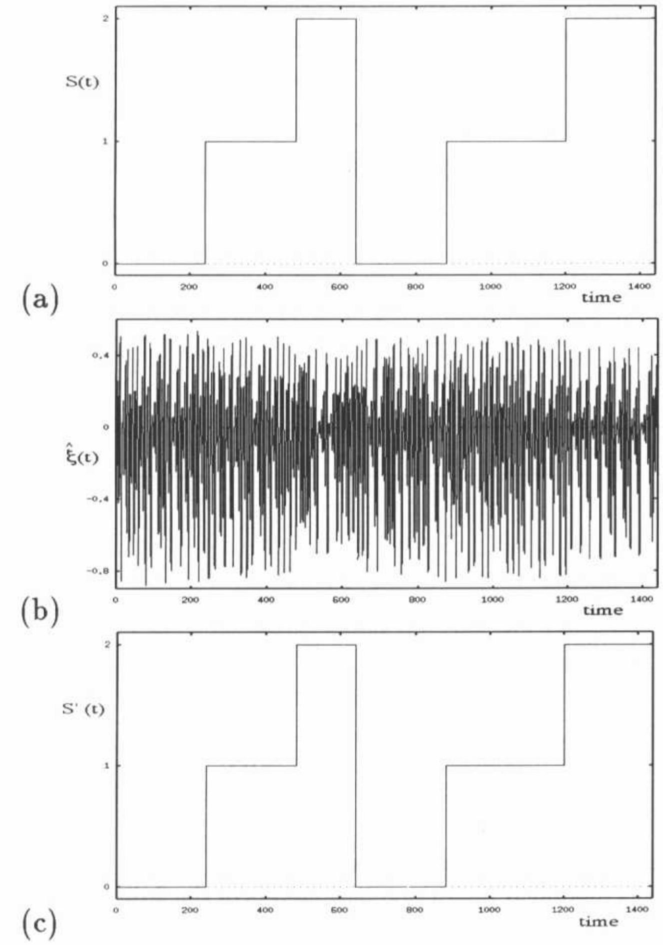


Figure 4.4: (a) Switch signal $s(t)$ of the bifurcation parameters (a,c) in the Rössler equation (4.31). (b) Average-filtered time series $\{\hat{\xi}_n\}$ from the switched Rössler equation. (c) Switch signal $s'(t)$ predicted by the present algorithm.

Figs. 4.4 and 4.5 show the results of the detection algorithm. The parameters of the algorithm are set as $(W, d, \tau, J, T, h, K, N_J, N_K) = (5, 3, 4, 6, 1200, 8, 2, 9988, 12)$. According to the principal component analysis of fig. 4.5 (a), cumulative significance ratio of $\Theta_2 > 0.98$ is obtained. Hence, we set the principal component parameters as $\Gamma_2 = (\gamma_1, \gamma_2)$. Fig. 4.5 (d) shows the locations of the principal parameter values $\{\Gamma_2(i)\}_{i=1,2,..,18}$ corresponding to the 18-windows of shortly divided trajectories $\{X_n(i)\}_{i=1,2,..,18}$ with a time interval of $S = 400$. The switch dynamics among the 3-clusters of distinctive points in the principal parameter space is clearly recognized. According to the LBG-clustering analysis of fig. 4.5 (c), it is shown that the optimal cluster number is $q_{opt} = 3$. As in fig. 4.4 (c), the switch signal $s'(t)$ of the LBG-cluster data accurately predicts the original signal $s(t)$.

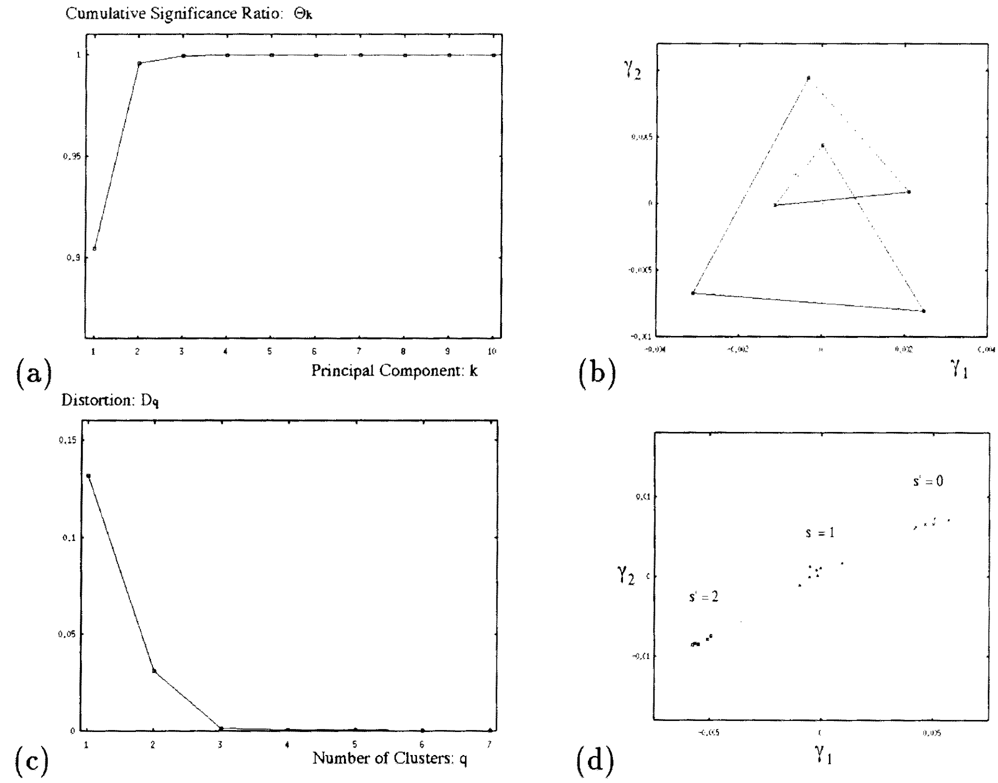


Figure 4.5: (a) Cumulative significance ratio $\{\Theta_k\}$. (b) Locations of the subsequence $\{\Omega(9988), .., \Omega(9999)\}$ of the nonlinear prediction parameters in the 2-dimensional principal space (γ_1, γ_2). (c) Distortion function D_q optimized by the LBG-clustering algorithm with a cluster number q. (d) Locations of the principal parameter values $\{\Gamma_2(i)\}_{i=1,2,..,18}$ corresponding to the windows of short-term trajectories $\{X_n(i)\}_{i=1,2,..,18}$. The principal parameters are classified into 3-groups, where the crosses indicate the points classified into "$s' = 0$," the triangles indicate the points classified into "$s' = 1$," and the squares indicate the points classified into "$s' = 2$."

4.3.3 Mackey-Glass equation

As a final example, we consider the Mackey-Glass difference-differential equation [137]:

$$\frac{d\eta_t}{dt} = a(s(t)) \; \frac{\eta_{t-17}}{1+\eta_{t-17}^{10}} \; - 0.1 \; \eta_t. \tag{4.34}$$

The bifurcation parameter a takes three values:

$$a(0) = 0.21, \;\; a(1) = 0.2, \;\; a(2) = 0.19, \tag{4.35}$$

according to the switch signal $s(t)$ of fig. 4.6 (a) and the corresponding chaotic time series is obtained as

$$\{\xi_n = \; \eta_{n\Delta t} + \nu_n \mid n = 1, 2, .., N_{data}\}, \tag{4.36}$$

where $\Delta t = 1.25$, $N_{data} = 6000$, $\nu_n \in N(0, 0.02)$, and the Mackey-Glass equation is numerically integrated by the fourth-order Runge-Kutta algorithm with a time step of 0.025.

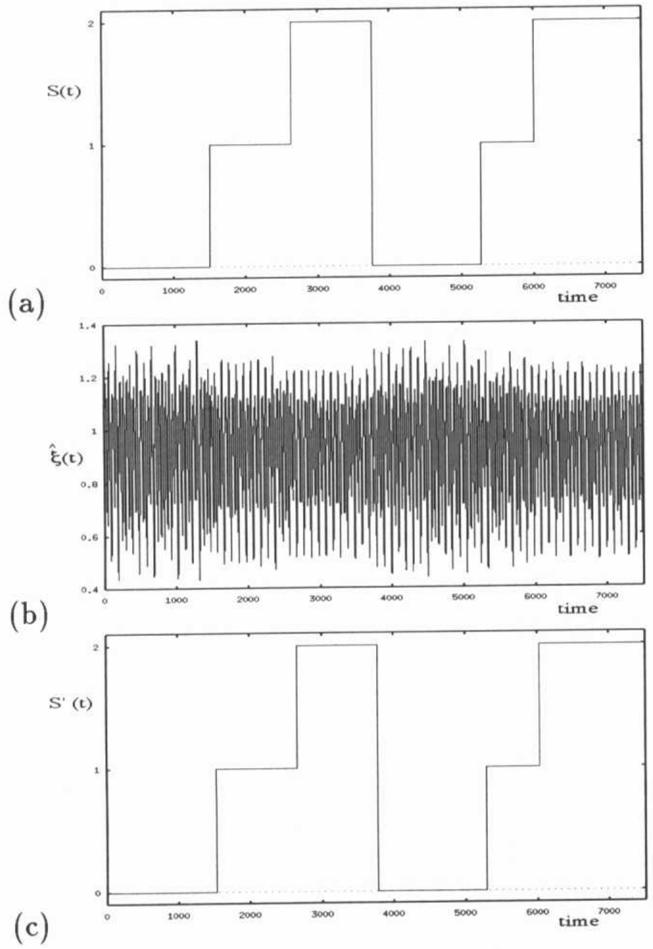


Figure 4.6: (a) Switch signal $s(t)$ of the bifurcation parameter a of the Mackey-Glass equation (4.34). (b) Average-filtered time series $\{\hat{\xi}_n\}$ from the switched Mackey-Glass equation. (c) Switch signal $s'(t)$ predicted by the present algorithm.

Figs. 4.6 and 4.7 show the results of the detection algorithm for the switch dynamics of the Mackey-Glass equation. The parameters of the algorithm are set as $(W, d, \tau, J, T, h, K, N_J, N_K) = (7, 4, 8, 6, 1000, 5, 2, 9988, 12)$. According to the principal component analysis of fig. 4.7 (a), cumulative significance ratio of $\Theta_1 > 0.98$ is obtained. Hence, we set the principal component parameters as $\Gamma_1 = (\gamma_1)$. Fig. 4.7 (d) shows the locations of the principal parameter values $\{\Gamma_1(i)\}_{i=1,2,..,20}$ corresponding to the 20-windows of shortly divided trajectories $\{X_n(i)\}_{i=1,2,..,20}$ with a time interval of $S = 300$. Switch dynamics among the 3-clusters of distinctive points in the principal parameter space is discernible. According to the LBG-clustering analysis of fig. 4.7 (c), optimal cluster number is correctly detected as $q_{opt} = 3$ and the original signal $s(t)$ can be accurately predicted by the LBG signal $s'(t)$ in fig. 4.6 (c).

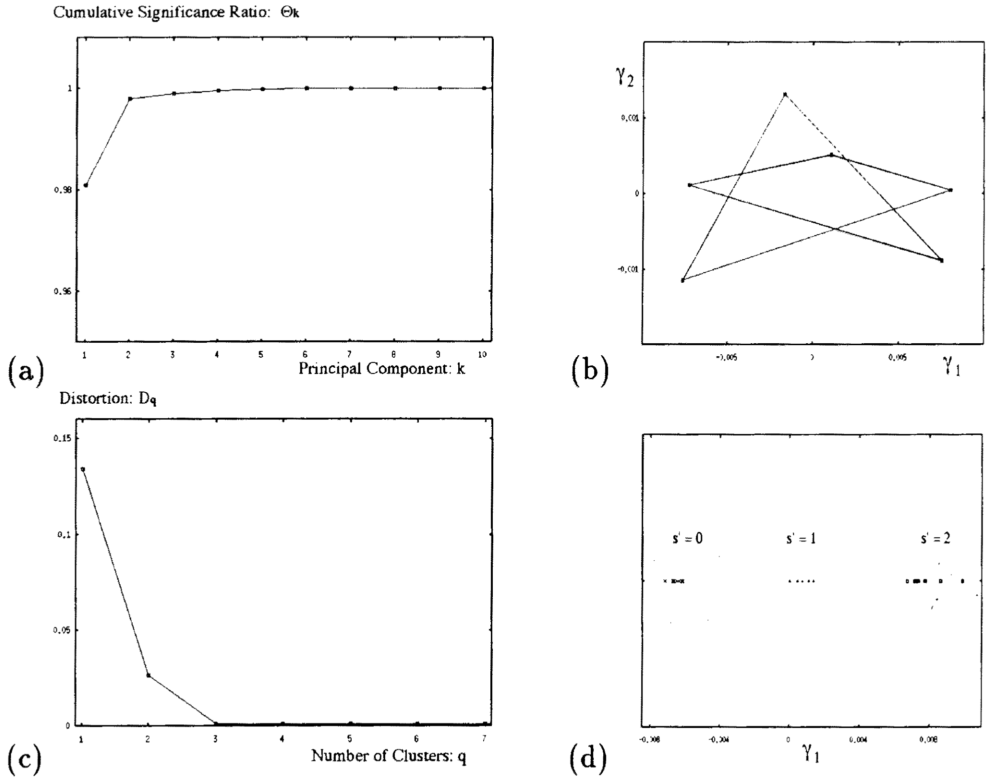


Figure 4.7: (a) Cumulative significance ratio $\{\Theta_k\}$. (b) Locations of the subsequence $\{\Omega(9988), .., \Omega(9999)\}$ of the nonlinear prediction parameters in the 1-dimensional principal space (γ_1). (c) Distortion function D_q optimized by the LBG-clustering algorithm with a cluster number q. (d) Locations of the principal parameter values $\{\Gamma_1(i)\}_{i=1,2,..,20}$ corresponding to the windows of short-term trajectories $\{X_n(i)\}_{i=1,2,..,20}$. The principal parameters are classified into 3-groups, where the crosses indicate the points classified into "$s' = 0$," the triangles indicate the points classified into "$s' = 1$," and the squares indicate the points classified into "$s' = 2$."

4.4 Conclusions and discussions

We have presented an algorithm for detecting switch dynamics in chaotic time series. By the switch dynamics, we mean that the bifurcation parameter values are occasionally changed in the chaotic time series. Using three chaotic dynamical systems, the Lorenz equations, the Rössler equations, and the Mackey-Glass equations, whose bifurcation parameters switch among two or three sets of slightly different parameter values, efficiency of the algorithm is shown. For the chaotic time series contaminated with observational noise, our algorithm has accurately detected the number of switching parameters as well as their switching points.

It should be noted that the present algorithm is based upon the characterization of windows of short-term chaotic time series in terms of the principal bifurcation parameters of nonlinear predictors. Performance of the algorithm to identify the number of switching parameters and their switching points is primarily dependent upon a reliable estimation of the principal bifurcation parameter values corresponding to each window of chaotic time series. Reliable estimation of the corresponding principal parameters becomes difficult when:

(a) The observational noise level is too high.

(b) The window length of chaotic time series is too short.

(c) The bifurcation parameters frequently make switches within a window of chaotic time series.

Exact number of the switching parameters may be accurately identified when the problems (a)-(c) are not so significant.

Limitation of the present algorithm against these problems will be studied in our future works. Applicability of the algorithm against higher-dimensional dynamical systems such as the spatio-temporal dynamical systems [40, 106] would be also considered in our further studies.

Chapter 5

Global bifurcation structure of chaotic neural networks and its application to traveling salesman problems

abstract: This Chapter studies global bifurcation structure of the chaotic neural networks applied to solve the traveling salesman problem (TSP). The bifurcation analysis clarifies the dynamical basis of the chaotic neuro-dynamics that itinerates a variety of network states associated with possible solutions of TSP and efficiently "searches" for the optimum or near-optimum solutions. By following the detailed merging process of čhaotic attractors via crises, we find that the crisis-induced intermittent switches among the ruins of the previous localized chaotic attractors underly the "chaotic search" for TSP solutions.

5.1 Introduction

The traveling salesman problem (TSP) is a classic and famous example of a combinatorial optimization problem which is hard to deal with. Computational time required to find an exactly optimum solution grows faster than any finite power of some appropriate measure of the problem size as long as $P \neq NP$ [69, 124, 164]. In order to cope with such hard problems, efficient approximate algorithms for finding a near-optimum solution within a reasonable computational time have been groped for. As one of such methods, this Chapter focuses on an intriguing optimization technique of TSP by chaotic dynamics [35, 154, 213] based on chaotic neural networks [5, 12].

In neural network approach to TSP, every possible solution of the TSP is mapped into a network of neurons with $(0,1)$-binary outputs [94]. Optimization by chaotic dynamics is to search for a better TSP solution by following a chaotic wandering orbit. By visiting a variety of network states which correspond to possible solutions of the TSP, chaotic dynamics continually searches for a better solution.

The remarkable "chaotic search" capability to TSP is demonstrated by Nozawa [154] who reported that for 94% of the random choices of initial conditions chaotic neural network with heuristically tuned values of system parameters finds optimum solution of 10-city TSP within 1000-iterative steps. Whereas the experimental studies demonstrate efficiency of the "chaotic search" to the optimization problem, we remark that they provide only preliminary results. Due to the complex dynamics of the chaotic neural network, the underlying dynamical mechanism of the "chaotic search" is not at all clear. Hence, the efficiency of the method can not be grasped theoretically.

The aim of the present Chapter is to clarify the underlying mechanism of the "chaotic search" and to re-examine the efficiency of the method to optimization problems. For our aim, we study global bifurcation structure of the chaotic neural networks applied to solve TSP. The bifurcation study not only clarifies the dynamical mechanism of the "chaotic search" but also provides a guideline for tuning the bifurcation parameter value which gives rise to the network dynamics with efficient "chaotic search."

The present Chapter is organized as follows: In Section 5.2, we review an experimental method for solving TSP by chaotic neural networks. In Sections 5.3 and 5.4, we study one-parameter bifurcation structure of the chaotic neural networks applied to 10- and 5-city instances of TSP. In particular, we follow in detail the merging process of chaotic attractors via crises [80] and find that the crisis-induced intermittent switching [79] underlies the "chaotic search" for TSP solutions. In Section 5.5, on the basis of the bifurcation studies of Sections 5.3 and 5.4, we discuss the global bifurcation structure of the chaotic neural network and the practical applicability of the "chaotic search" to the optimization problem.

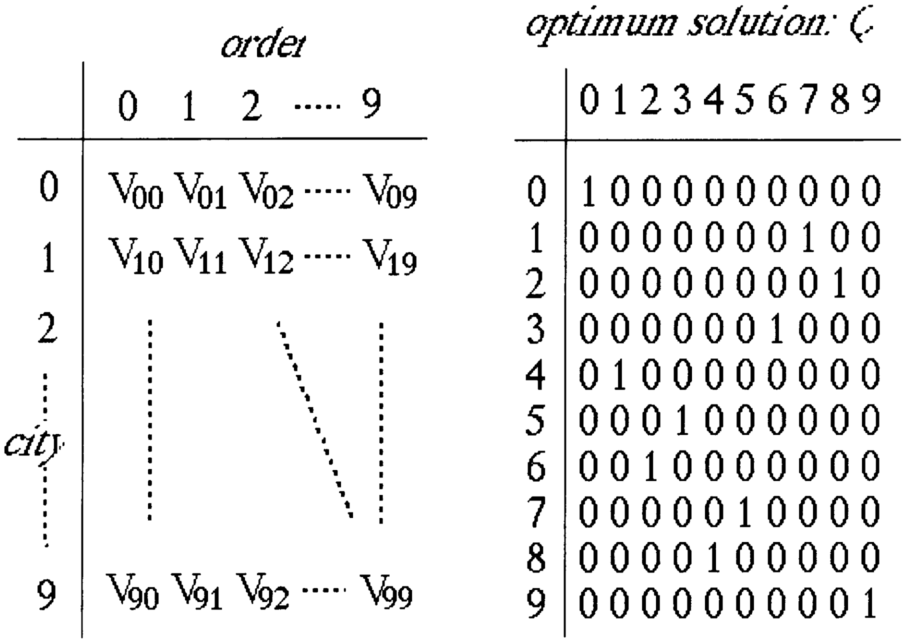
5.2 Experiment on solving TSP by chaotic neural network

This Section reviews the experimental method for solving TSP by chaotic neural networks [154]. First, the Hopfield-Tank neural network is introduced to solve TSP. Second, chaotic neural network is derived by the Euler's discretization of the continuous-time Hopfield-Tank neural network. Third, a technique for observing a set of temporal firing rates of neurons as possible TSP solution is described.

5.2.1 Hopfield-Tank neural network approach to TSP

Consider an N-city traveling salesman problem (TSP): Given an $N \times N$ symmetric matrix (d_{ij}) of distances between a set of N cities $(i, j = 0, 1, .., N-1)$, find a minimum-length tour that visits each city exactly once.

A solution for the TSP can be described in terms of a $N \times N$ matrix V with $(0,1)$-binary elements $\{V_{ik} = 0,1 \mid i,k = 0,1,..,N-1\}$. Any complete tour can be represented by denoting $V_{ik} = 1$ if city i is in position k in the tour and $V_{ik} = 0$ otherwise, where the position k stands for a visiting order in the tour (see fig. 5.1).


order
city
optimum solution: Q

	0	1	2	.....	9
0	V_{00}	V_{01}	V_{02}	.....	V_{09}
1	V_{10}	V_{11}	V_{12}	.....	V_{19}
2	:				:
:	:				:
9	V_{90}	V_{91}	V_{92}	.....	V_{99}

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0
4	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1

Figure 5.1: $N \times N$-element binary code $V = \{V_{ik} \mid i,k = 0,1,..,N-1\}$ as a solution to TSP. Shown right is the optimum solution Q of the 10-city TSP studied in this Chapter. The solution represents a tour in which city 0 is the first city to be visited, city 4 the second, city 6 the third, and so forth.

Preference for the matrix V as a solution to the TSP can be measured by the following cost function:

$$E(V) = \frac{A}{2}\sum_{i=0}^{N-1}\{\sum_{k=0}^{N-1} V_{ik} - 1\}^2 + \frac{A}{2}\sum_{k=0}^{N-1}\{\sum_{i=0}^{N-1} V_{ik} - 1\}^2 + \frac{B}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} d_{ij}V_{ik}\{V_{j\,k+1} + V_{j\,k-1}\}. \quad (5.1)$$

Whereas the first two terms represent the constraint terms to satisfy the feasibility condition of TSP, the third term represents a total path length of a complete tour of feasible TSP solution. The solution matrix V associated with a low cost value represents a candidate of a good TSP solution, since the low cost solution may satisfy the constraints and provide a short length tour.

On the basis of the representation of a TSP solution by the $N \times N$ matrix V, the Hopfield-Tank neural network is designed as

$$R\,(\frac{d}{dt}u_{ik}) = -\,u_{ik} + \sum_{j=0}^{N-1}\sum_{l=0}^{N-1} T_{ik,jl}V_{jl} + I_{ik}, \qquad V_{ik} = \sigma(u_{ik}), \tag{5.2}$$

where u_{ik} stands for the internal state variable of the (i,k)-neuron $(0 \le i, k \le N-1)$, R stands for a time constant parameter, and $\sigma(x) = 0.5 + 0.5\tanh(x/\beta)$. The synaptic connections $T_{ik,jl}$ are given by

$$\begin{aligned} T_{ik,jl} &= -A(\delta_{ij}(1-\delta_{kl}) + \delta_{kl}(1-\delta_{ij})) - Bd_{ij}(\delta_{l\,k+1} + \delta_{l\,k-1}), \; (i,k) \ne (j,l), & (5.3)\\ T_{ik,ik} &= -2A, & (5.4)\\ I_{ik} &= 2A, & (5.5) \end{aligned}$$

for $0 \le\ i,k,j,l\ \le N-1$.

The Hopfield-Tank neural network of eq. (5.2) has potential capability to solve the TSP because:

(i) The underlying Lyapunov function is defined as

$$H(V) = E(V) + \sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\int_0^{V_{ik}} \sigma^{-1}(V)dV \tag{5.6}$$

which is, for small β, nearly equal to the TSP cost function of eq. (5.1), since $\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\int_0^{V_{ik}} \sigma^{-1}(V)dV \to 0\ (\beta \to 0)$.

(ii) Since, for small β, the minimal states of the Lyapunov function appear nearly with $(0,1)$-binary outputs $\{V_{ik} = 0,1 \mid i,k = 0,1,..,N-1\}$, final equilibrium states of the network (5.2) may provide feasible TSP solutions with short total-path length.

5.2.2 Chaotic neural network approach to TSP

Nevertheless, practical applicability of the Hopfield-Tank neural network to optimization problems is crucially limited by the following problems.

(i) Choosing the parameter values for (A, B) which controls the strength of the constraint terms against the tour-length term in eq. (5.1) is quite difficult. In fact, it is reported in [86, 210] that appropriate parameter values for (A, B) which properly locate feasible TSP solutions into local minima of the Lyapunov function $H(V)$ lie in a small restricted space. Most choices of (A, B) fail to locate feasible TSP solutions as the local minima and instead produce a large number of local minima which do not satisfy the constraints of eq. (5.1).

(ii) Even with an appropriate selection of the parameter values (A, B), the Lyapunov function $H(V)$ still has a large number of feasible solutions from long-length tours to nearly shortest tours. Due to the existence of the large number of long length tours which are far from the optimum tour, the network is frequently trapped in local minima with such bad solutions, depending on the choice of initial conditions.

For (i), various techniques are investigated for determining good parameter values for (A, B) [13, 210]. For (ii), stochastic dynamics is usually introduced to escape from the local minima with long length tours [2, 114]. In chaotic neural network approach to TSP, the network escapes from the local minima by chaotic non-equilibrium dynamics [12]. The chaotic neural network for the combinatorial optimization problem is formulated as follows.

First, parameter values for the synaptic connections (5.3)-(5.5) are slightly modified as

$$
\begin{aligned}
T_{ik,jl} &= -A(\delta_{ij}(1-\delta_{kl}) + \delta_{kl}(1-\delta_{ij})) - Bd_{ij}(\delta_{l\,k+1} + \delta_{l\,k-1}), \ (i,k) \neq (j,l), && (5.7)\\
T_{ik,ik} &= -2\omega A, && (5.8)\\
I_{ik} &= 2\alpha A, && (5.9)
\end{aligned}
$$

where $0 \leq \ i, k, j, l \ \leq N-1$, α is a control parameter for excitation level of neurons, and ω is a parameter for adjusting the negative self-feedback [35, 154] or the refractory effect [12].

Second, with an affine transformation $u_{ik} = \sum_{j=0}^{N-1}\sum_{l=0}^{N-1} T_{ik,jl}\, p_{jl} + I_{ik}$, the Hopfield-Tank neural network (5.2) is transformed into the following form

$$
R\,(\frac{d}{dt}p_{ik}) = -\,p_{ik} + V_{ik}, \qquad V_{ik} = \sigma(\sum_{j=0}^{N-1}\sum_{l=0}^{N-1} T_{ik,jl}\, p_{jl} + I_{ik}). \tag{5.10}
$$

If the connection matrix T is invertible, it is proven by Pineda [162] that the attractor structures of the two equations (5.2) and (5.10) are identical. The chaotic neural network is then derived by the Euler's discretization of the continuous-time model (5.10) as

$$\begin{aligned} p_{ik}(n+1) &= p_{ik}(n) + \frac{\Delta t}{R}\ (-p_{ik}(n) + \sigma(\sum_{j=0}^{N-1}\sum_{l=0}^{N-1} T_{ik,jl}\, p_{jl}(n) + I_{ik})\ ) \\ &= r\ p_{ik}(n) + (1-r)\ \sigma(\sum_{j=0}^{N-1}\sum_{l=0}^{N-1} T_{ik,jl}\, p_{jl}(n) + I_{ik}), \end{aligned} \tag{5.11}$$

where Δt is the time step of the Euler's discretization and $r = 1 - (\Delta t/R)$. The model is equivalent to a single internal state version of the chaotic neural network [12, 154].

With a set of well selected values of the parameters $(A, B, \omega, \alpha, \beta)$, the chaotic neural network exhibits chaotic dynamics which "searches" for TSP solutions. The search procedure for TSP solutions can be observed by calculating the temporal firing rates of neurons as follows.

At every time step n, first, we compute short-term averaged firing rates $\rho(n) = \{\rho_{ik}(n) = (1/w)\sum_{j=0}^{w-1} p_{ik}(n-j) \mid i,k = 0,1,..,N-1\}$ of neurons with an averaging duration w. The temporal network firing state $\rho(n)$ is then encoded into an $N \times N$-element binary code $J(n) = \{J_{ik}(n) \mid i,k = 0,1,..,N-1\}$ as

$$J_{ik}(n) = 1[\rho_{ik}(n) - \rho^*(n)] \tag{5.12}$$

where $1[x] = 1$ $(x \geq 0)$, $1[x] = 0$ $(x < 0)$, and $\rho^*(n)$ is the N-th largest value among $\{\rho_{ik}(n) \mid i,k = 0,1,..,N-1\}$.

By wandering around a variety of $N \times N$-element binary codes $\{J(n) \mid n = 0,1,..\}$ with possible TSP solutions, chaotic dynamics "searches" for a better TSP solution. In the next Section, we study the dynamical basis of the "chaotic search."

5.3 One-parameter bifurcation of the chaotic neural network applied to solve 10-city TSP

The experiment of applying the chaotic neural network to solve the Hopfield-Tank's 10-city TSP is carried out by Nozawa [154] with a set of fixed parameter values. By taking one of the parameters as a bifurcation parameter, we study one-parameter bifurcation structure of the chaotic neural network and clarify the dynamical mechanism of the Nozawa experiment.

First, we show that the chaotic neural network of eq. (5.11) has symmetry which characterizes the global bifurcation structure. Second, a simple coding rule which enables to map every attractor of the dynamical system to possible TSP solution is introduced. Third, one-parameter bifurcation structure of the chaotic neural network is studied.

5.3.1 Symmetry in dynamical systems

Consider a set of transformations $G = \{\eta^l \circ \gamma^m \mid l = 0,1,\ m = 0,..,N-1\}$ with $N \times N$-dimensional linear transformations of γ and η defined as

$$\gamma : p_{ik} \mapsto p_{i\,(k+1)\mathrm{mod}N} \qquad \text{for } 0 \le i,k \le N-1, \tag{5.13}$$

$$\eta : p_{ik} \mapsto p_{i\,(N-k)\mathrm{mod}N} \qquad \text{for } 0 \le i,k \le N-1, \tag{5.14}$$

where $\circ$ denotes composition and $\gamma^m = \underbrace{\gamma \circ \cdots \circ \gamma}_{m \text{ times}}$.

By using an $N \times N$-dimensional mapping $f : R^{N\times N} \to R^{N\times N}$, let us denote the network dynamics of eq. (5.11) by $p(n+1) = f(p(n))$. Since $f \circ g = g \circ f$ for any $g \in G$ (see Appendix A-2), the dynamical system is invariant under the operation by any $g \in G$. Hence the set of transformations G provides symmetry of the dynamical system of eq. (5.11).

The idea of the symmetry is important, because if $O = \{p(n) \mid n = 0,1,..\}$ is any solution of the dynamical system, then so is $g(O) = \{g(p(n)) \mid n = 0,1,..\}$ for all $g \in G$. In particular, if O is an attractor, then so is $g(O)$ for all $g \in G$. We then say that $g(O)$ is conjugate to O. With respect to the system symmetry, we can also characterize an attractor O by the following symmetry group [76] :

$$\Delta_O = \{g \in G \mid g(O) = O\}. \tag{5.15}$$

In the sense that the above group measures the degree of symmetry of O, we say that an attractor O is a Δ_O-symmetric attractor. If $\Delta_O = \emptyset$, we then say that an attractor O is an asymmetric attractor.

5.3.2 Coding of attractors

In the study of high-dimensional dynamical systems which give rise to multi-stability of many attractors, introduction of a simple coding of attractors is useful for a systematic analysis of the system. For instance, in the study of global couple map [107, 108], large number of multi-stable attractors are coded by the clustering conditions, while, in the study of optical system [99], attractors are coded by the branching order of the harmonic bifurcations. In the present analysis, every attractor in the neural dynamical system is coded into possible TSP solution as follows.

First, a set of long-term average firing rates $\{\bar{\rho}_{ik} = \lim_{T\to\infty}(1/T)\sum_{n=0}^{T-1} p_{ik}(n) \mid i,k = 0,1,..,N-1\}$ is measured on an attractor $\{p(n) \mid n = 0,1,..\}$. The attractor is then encoded into an $N \times N$-element binary code $\bar{J} = \{\bar{J}_{ik} \mid i,k = 0,1,..,N-1\}$ defined as

$$\bar{J}_{ik} = 1[\bar{\rho}_{ik} - \bar{\rho}^*] \tag{5.16}$$

where $\bar{\rho}^*$ is the N-th largest value among $\{\bar{\rho}_{ik} \mid i,k = 0,1,..,N-1\}$.

As is shown in Appendix A-3 , a set of conjugate attractors $\{\eta^l \circ \gamma^m(O) \mid l = 0,1,\ m = 0,..,N-1\}$ are coded into an equivalent TSP solution by this coding, if $O = \{p(n) \mid n = 0,1,..\}$ is an asymmetric attractor.

5.3.3 One-parameter bifurcation of the chaotic neural network

Taking r as a bifurcation parameter, let us study one-parameter bifurcation structure of the chaotic neural network applied to solve the Hopfield-Tank's 10-city TSP [94, 210] (see fig. 5.2). The values of the parameters are fixed as $(A, B, \omega, \alpha, \beta) = (1.0, 1.0, 0.75, 0.05, 0.018)$ so that the experimental situation of Nozawa [154] is reproduced at $r = 0.70$.

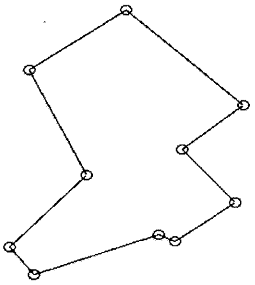


Figure 5.2: Locations of the Hopfield-Tank's 10-city TSP, which are given in 2-dimensional coordinates as (0.2439, 0.1463), (0.8488, 0.3609), (0.6683, 0.2536), (0.6878, 0.5219), (0.1707, 0.2293), (0.2293, 0.7610), (0.4000, 0.4439), (0.8732, 0.6536), (0.5171, 0.9414), (0.6195, 0.2634) [94, 210]. The example tour represents the optimum solution.

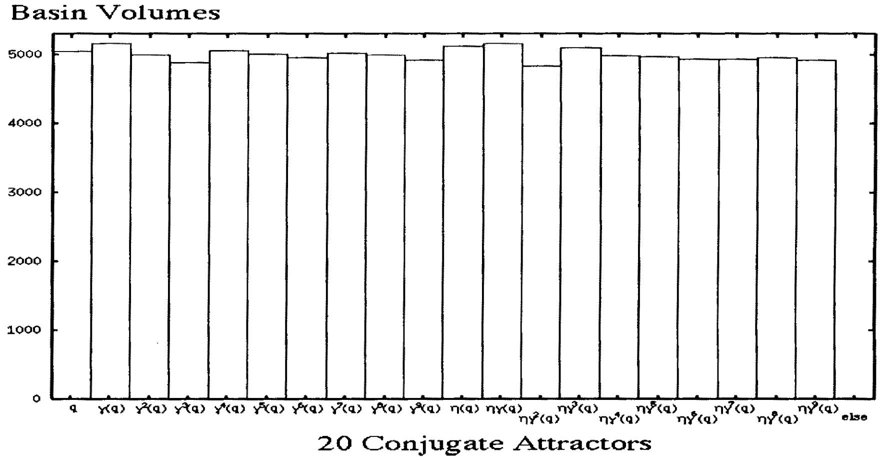


Figure 5.3: 100000 samples of random initial conditions $p(0) \in [0,1]^{N\times N}$ are classified into basins of 20 conjugate fixed points $\{\eta^l \circ \gamma^m(q) \mid l = 0, 1,\ m = 0, .., 9\}$ (q ($\in R^{N\times N}$) denotes a basic fixed point) or other attractors in the chaotic neural network with $r = 0.99$. The abscissa indicates the 20 conjugate fixed points and the other attractors and the ordinate indicates the number of initial conditions included in the basin of each attractor.

As is discussed in Section 5.2, the chaotic neural network approaches to the continuous-time Hopfield-Tank neural network when $r \to 1.0$. Hence, parameter values close to $r = 1$ are expected to give rise to local minimum solutions of the Hopfield-Tank network. First, by setting the parameter value to $r = 0.99$, we find the local minimum solutions by carpet-bombing algorithm [157]. As is shown in fig. 5.3, from 100000 samples of random initial conditions which are uniformly distributed over $p(0) \in [0,1]^{N\times N}$, only a set of $2N$ conjugate local minima $\{\eta^l \circ \gamma^m(q) \mid l = 0,1,\ m = 0,..,N-1\}$ are found, where $q\ (\in R^{N\times N})$ denotes a basic local minimum.

By giving this q as the initial condition at $r = 0.89$ and decreasing the bifurcation parameter from $r = 0.89$ to $r = 0.65$, the bifurcation diagram is drawn by observing the $(0,0)$-neuron state $p_{00}(n)$ in figs. 5.4 (a) and (b). The successively observed attractors are coded into possible TSP solutions $\bar{J}$ and the corresponding cost function values $E(\bar{J})$ defined by eq. (5.1) are plotted in fig. 5.4 (c). We remark that until about $r \cong 0.725$ the cost function curve constantly holds the value corresponding to the optimum-tour length of the TSP. Namely, all the observed attractors until $r \cong 0.725$ are coded as the optimum solution. We denote this optimum solution by code Q. Fig. 5.4 (d) shows the Lyapunov dimension D_L [110] of the attractors observed in the bifurcation diagram.

Parameter region of $\mathbf{0.89 > r > 0.725}$: In the first stage of this parameter region, the local minimum q continually exists until $r \cong 0.87$ and then bifurcates into chaotic attractor through period-doubling bifurcation route to chaos [55] (see fig. 5.4 (b)). With a decrease in the bifurcation parameter, the chaotic attractor continually increases in size. Whereas the repeated occurrence of saddle-node bifurcations gives rise to pairs of stable and unstable periodic solutions generating periodic windows in the bifurcation diagram, the stable periodic attractors also bifurcate into chaotic attractors which merge with the original chaotic attractor via an interior crisis [80]. Fig. 5.5 illustrates how the unstable periodic orbits which are born from the period-doubling bifurcations of the local minimum q underlie the structure of the chaotic attractors.

Let us denote the successively observed attractors in fig. 5.4 (a) by O_r, which depends upon the value of the bifurcation parameter r. Then the system symmetry implies that a conjugate attractor $\eta^l \circ \gamma^m(O_r)$ also undergoes a series of same bifurcations with O_r in this parameter region. Note that a conjugate attractor $\eta^l \circ \gamma^m(O_r)$ described above is continually coded as $\eta^l \circ \gamma^m(Q)$, which represents the optimum solution (see Appendix A-3). Interestingly, at most stages of their successive bifurcations, the $2N$ conjugate attractors $\{\eta^l \circ \gamma^m(O_r) \mid l = 0,1,\ m = 0,..,N-1\}$ are the only observable attractors of the system. In practice, for systems with $r = 0.89, 0.85, 0.78, 0.75$, our numerical experiments show that all the initial conditions on a hyper-surface: $(p_{00}, p_{01}) \in [0,1] \times [0,1], p_{02} = p_{03} = \cdots = p_{99} = 0$ can be classified into basins of the conjugate attractors $\{\eta^l \circ \gamma^m(O_r) \mid l = 0,1,\ m = 0,..,N-1\}$ (see figs. 5.6 (a)-(f)).

Parameter region of $\mathbf{0.74 > r}$: As is shown in fig. 5.4 (d), the $2N$ conjugate chaotic attractors increase in their size as the bifurcation parameter is further decreased. When the parameter reaches a value of about $r \cong 0.725$, the thick chaotic band suddenly disappears from the bifurcation diagram of fig. 5.4 (a). It seems that the $2N$ conjugate chaotic attractors collide with each other and merge as a single attractor via symmetry-increasing crises [37, 38].

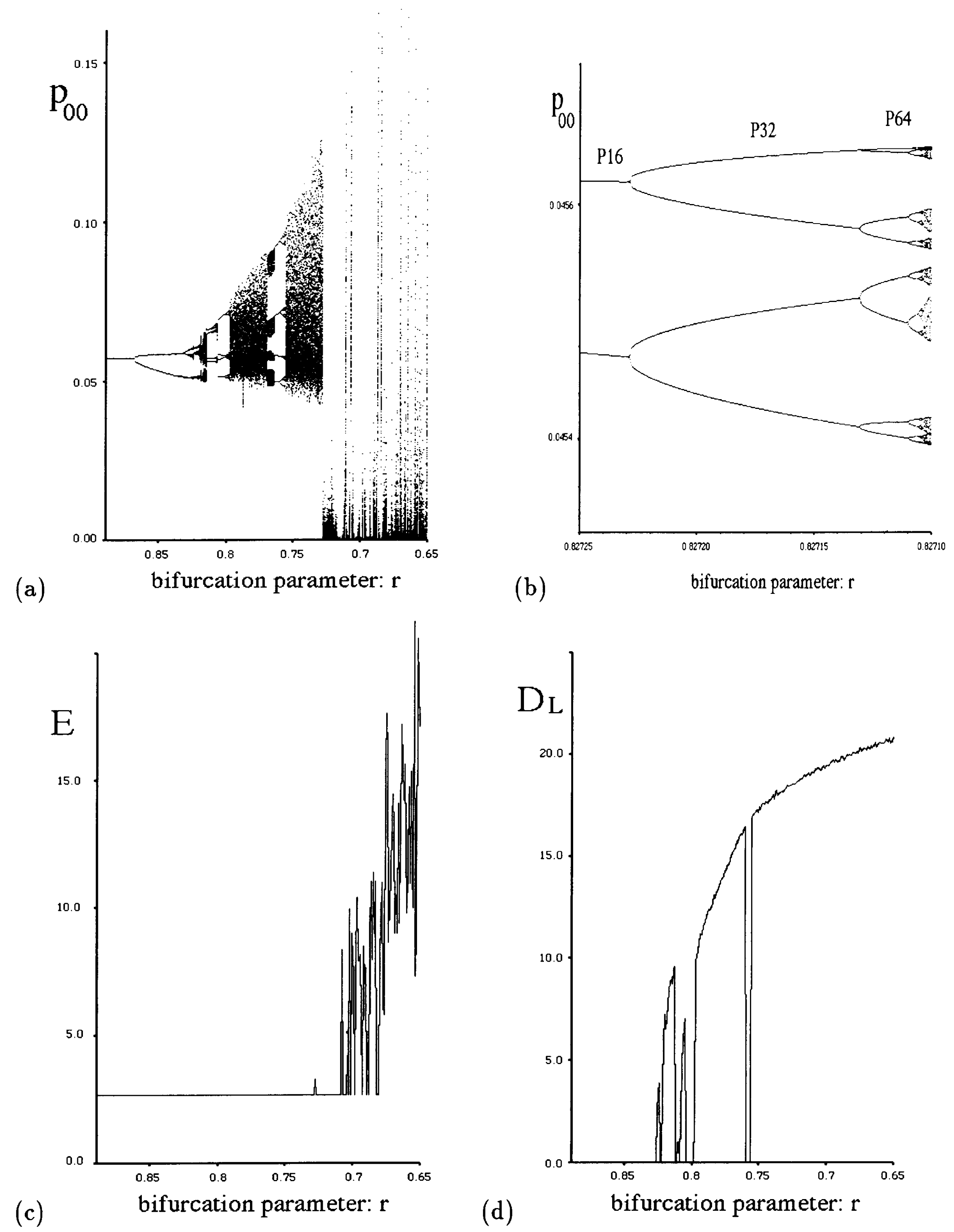


Figure 5.4: A one-parameter bifurcation diagram of the chaotic neural network for the 10-city TSP. The bifurcation parameter value r is decreased from 0.89 to 0.65. (a) A one-parameter bifurcation diagram observed from the $(0,0)$-neuron state $p_{00}(n)$. (b) Enlargement of the bifurcation diagram of (a). The period-16 attractor bifurcates into period-32, period-64, and so forth into a chaotic attractor. (c) The cost function values $E(\bar{J}(O_r))$ defined by eq. (5.1) for the successively observed attractors O_r in (a). (d) The Lyapunov dimersion D_L of the successively observed attractors O_r in (a).

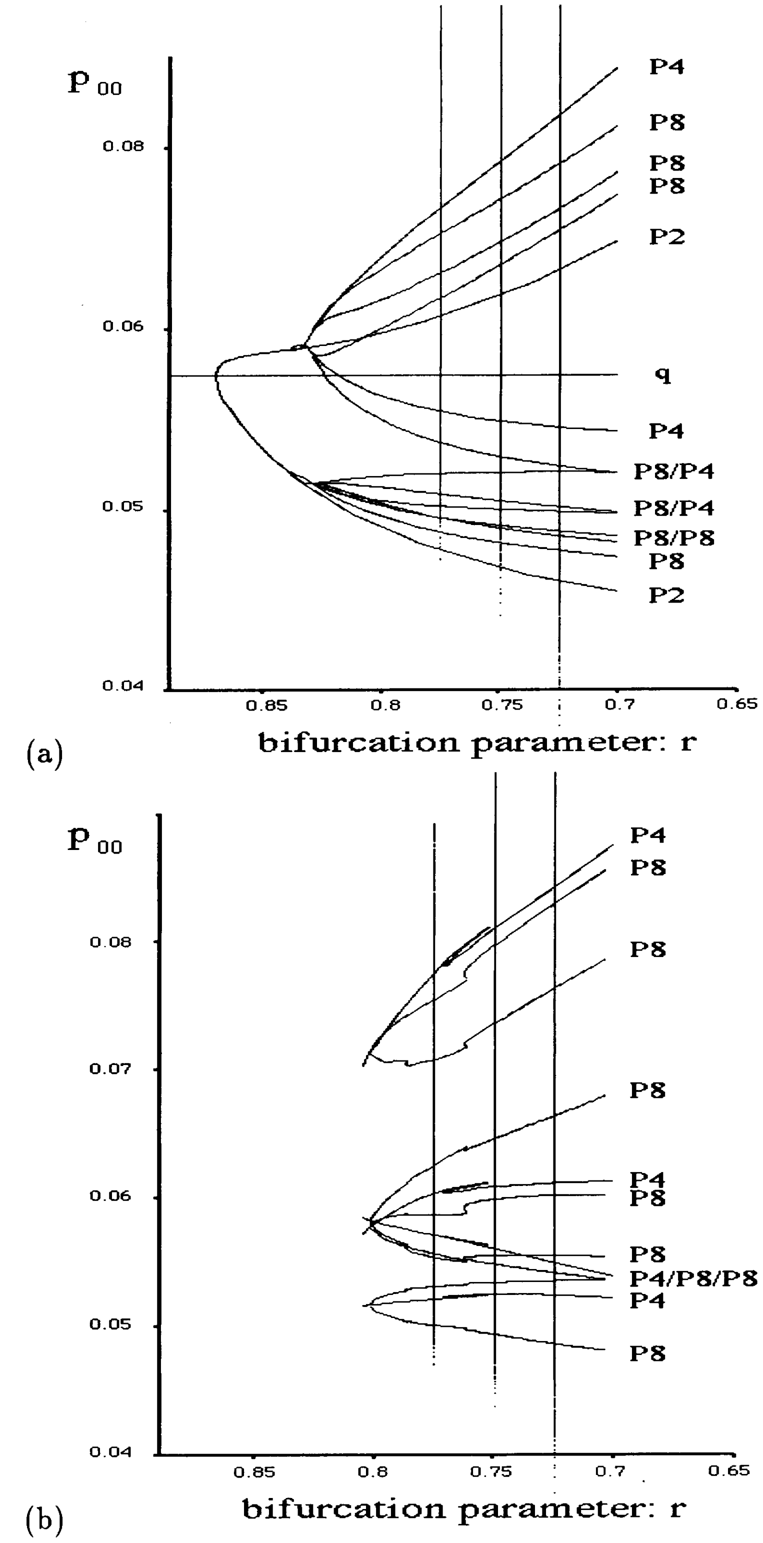


Figure 5.5: Period-2, 4, and 8 orbits born from the period-doubling bifurcations of the local minimum q. The figure displays how these unstable periodic orbits underly the structure of the chaotic attractors at $r = 0.775,\ 0.750,\ 0.725$.

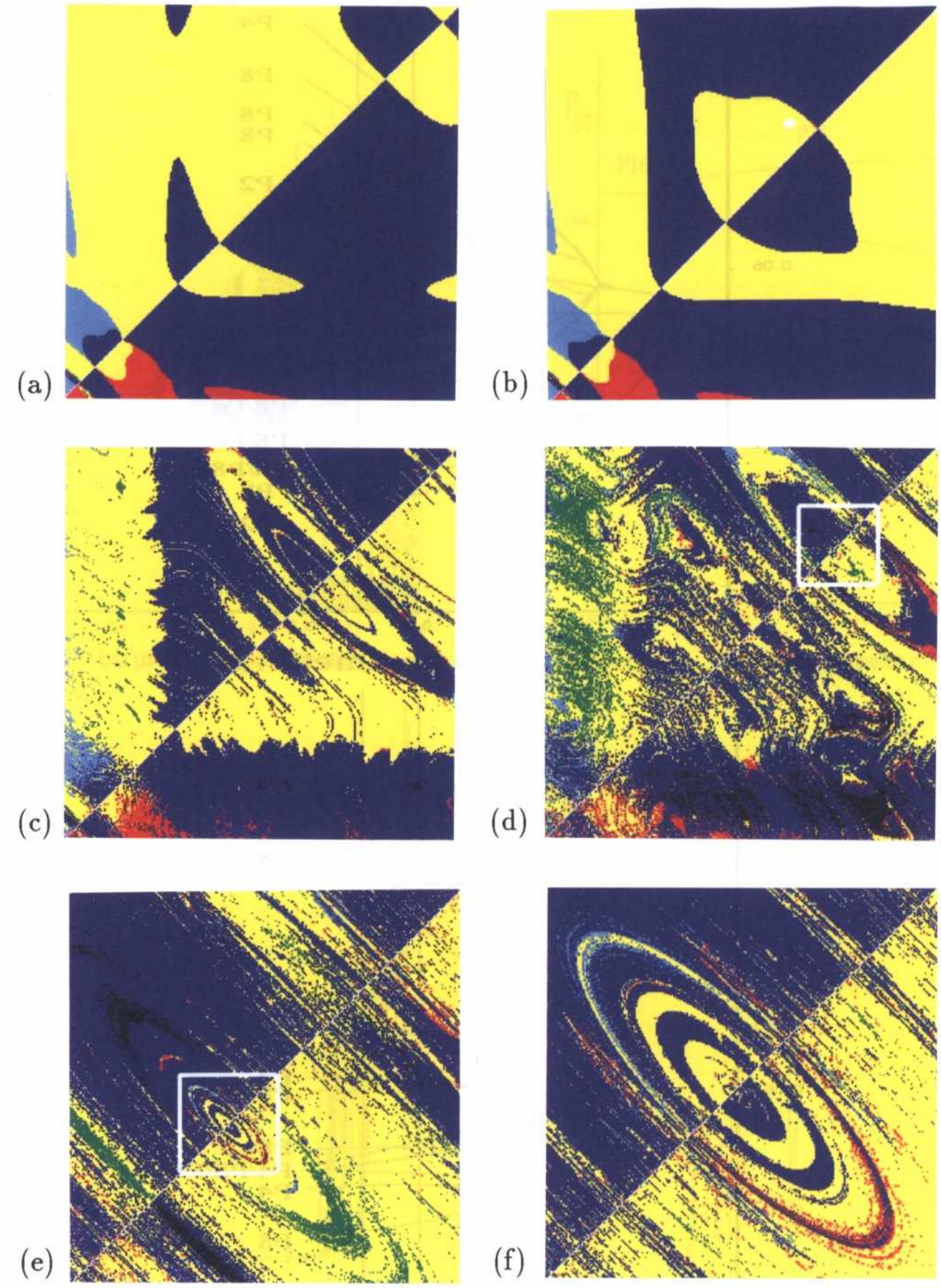


Figure 5.6: On the hyper-surface with $(p_{00}, p_{01}) \in [0,1] \times [0,1], p_{02} = p_{03} = \cdots = p_{99} = 0$, basin portraits of the conjugate attractors $\{\eta^l \circ \gamma^m(O_r) \mid l = 0,1, \;\; m = 0,..,N-1\}$ are drawn for four systems: (a) $r = 0.89$, (b) $r = 0.85$, (c) $r = 0.78$, and (d) $r = 0.75$. The abscissa indicates $p_{00} \in [0,1]$ and the ordinate indicates $p_{01} \in [0,1]$. While (e) shows an enlargement of the square part: $(p_{00}, p_{01}) \in [0.65, 0.85] \times [0.65, 0.85]$ of (d), (f) shows a further enlargement of the square part: $(p_{00}, p_{01}) \in [0.707, 0.757] \times [0.707, 0.757]$ of (e). With respect to the basic attractors O_r observed in the bifurcation diagram of fig. 5.4 (a), basins of O_r, $\gamma^7(O_r)$, and $\eta \circ \gamma^4(O_r)$ are colored by yellow, basins of $\gamma(O_r)$, $\gamma^8(O_r)$, and $\eta \circ \gamma^5(O_r)$ are colored by red, basins of $\gamma^2(O_r)$, $\gamma^9(O_r)$, and $\eta \circ \gamma^6(O_r)$ are colored by green, basins of $\gamma^3(O_r)$, $\eta(O_r)$, and $\eta \circ \gamma^7(O_r)$ are colored by black, basins of $\gamma^4(O_r)$, $\eta \circ \gamma(O_r)$, and $\eta \circ \gamma^8(O_r)$ are colored by blue, basins of $\gamma^5(O_r)$, $\eta \circ \gamma^2(O_r)$, $\eta \circ \gamma^9(O_r)$ are colored by skyblue, and basins of $\gamma^6(O_r)$ and $\eta \circ \gamma^3(O_r)$ are colored by purple.

5.3.4 Switching among previous localized chaotic attractors and chaotic search for TSP solutions

Let us study the network dynamics after the mergers of the $2N$ conjugate chaotic attractors via crises. The crises may give rise to intermittent switching among the previous localized chaotic attractors [79]. By following the details of the switches, we find that the crisis-induced intermittent switches are the dynamical bases of the "chaotic search" for TSP solutions.

Recalling that the basic attractors O_r observed in the bifurcation diagram of fig. 5.4 (a) have been continually coded as the optimum solution Q, we denote the set of the conjugate chaotic attractors $\{\eta^l \circ \gamma^m(O_r) \mid l = 0, 1,\ m = 0, .., N-1\}$ just before the crises by $\{\eta^l \circ \gamma^m(Q) \mid l = 0, 1,\ m = 0, .., N-1\}$. Then, we can follow the details of the switches among the ruins of the previous localized chaotic attractors $\{\eta^l \circ \gamma^m(Q)\}$ by calculating the temporal network firing state $J(n)$ defined by eq. (5.12) with an averaging duration w. For instance, if $J(n) = \eta^l \circ \gamma^m(Q)$, then we may consider that the temporal network firing state at the time step n is calculated over the duration during which the system is around the (l, m)-previous localized chaotic attractor $\eta^l \circ \gamma^m(Q)$.

Fig. 5.7 (a) shows a sequence $\{J(n) \mid n = 0, 1, ..\}$ of the temporal network dynamics with $r = 0.73$. The dynamic behavior is from a given random initial condition and the averaging duration is set to $w = 500$. Starting from Q, cyclic switches among the previous localized chaotic attractors as

$$\text{“}Q \rightharpoonup \gamma(Q) \rightharpoonup \cdots \rightharpoonup \gamma^m(Q) \rightharpoonup \gamma^{m+1}(Q) \rightharpoonup \cdots \rightharpoonup \gamma^9(Q) \rightharpoonup Q\text{”} \tag{5.17}$$

are recognized. From a different initial condition, cyclic switches as

$$\text{“}\eta(Q) \rightharpoonup \eta \circ \gamma(Q) \rightharpoonup \cdots \rightharpoonup \eta \circ \gamma^m(Q) \rightharpoonup \eta \circ \gamma^{m+1}(Q) \rightharpoonup \cdots \rightharpoonup \eta \circ \gamma^9(Q) \rightharpoonup \eta(Q)\text{”} \tag{5.18}$$

are also observed. This implies that a first crisis gives birth to two attractors (5.17) and (5.18), which are conjugate with respect to η.

Fig. 5.8 shows an average of the residence time τ_{av} [79], in which the network stays in one of the previous localized chaotic attractors $\{\eta^l \circ \gamma^m(Q) \mid l = 0, 1,\ m = 0, .., N-1\}$. At the initial stage of the crisis, switching rarely occurs and the residence time τ_{av} is inordinately long. As is explained in detail below, with a decrease in the bifurcation parameter, the previous localized chaotic attractors get more tightly connected to each other. This shortens the residence time τ_{av} in fig. 5.8.

For $0.73 \leq r \leq 0.737$, the system exhibits cyclic switching as (5.17) and (5.18). As the crisis proceeds for $0.66 < r < 0.73$, reversible switches as

$$\text{“}Q \rightleftharpoons \gamma(Q) \rightleftharpoons \cdots \rightleftharpoons \gamma^m(Q) \rightleftharpoons \gamma^{m+1}(Q) \rightleftharpoons \cdots \rightleftharpoons \gamma^9(Q) \rightleftharpoons Q\text{”} \tag{5.19}$$

and

$$\text{“}\eta(Q) \rightleftharpoons \eta \circ \gamma(Q) \rightleftharpoons \cdots \rightleftharpoons \eta \circ \gamma^m(Q) \rightleftharpoons \eta \circ \gamma^{m+1}(Q) \rightleftharpoons \cdots \rightleftharpoons \eta \circ \gamma^9(Q) \rightleftharpoons \eta(Q)\text{”} \tag{5.20}$$

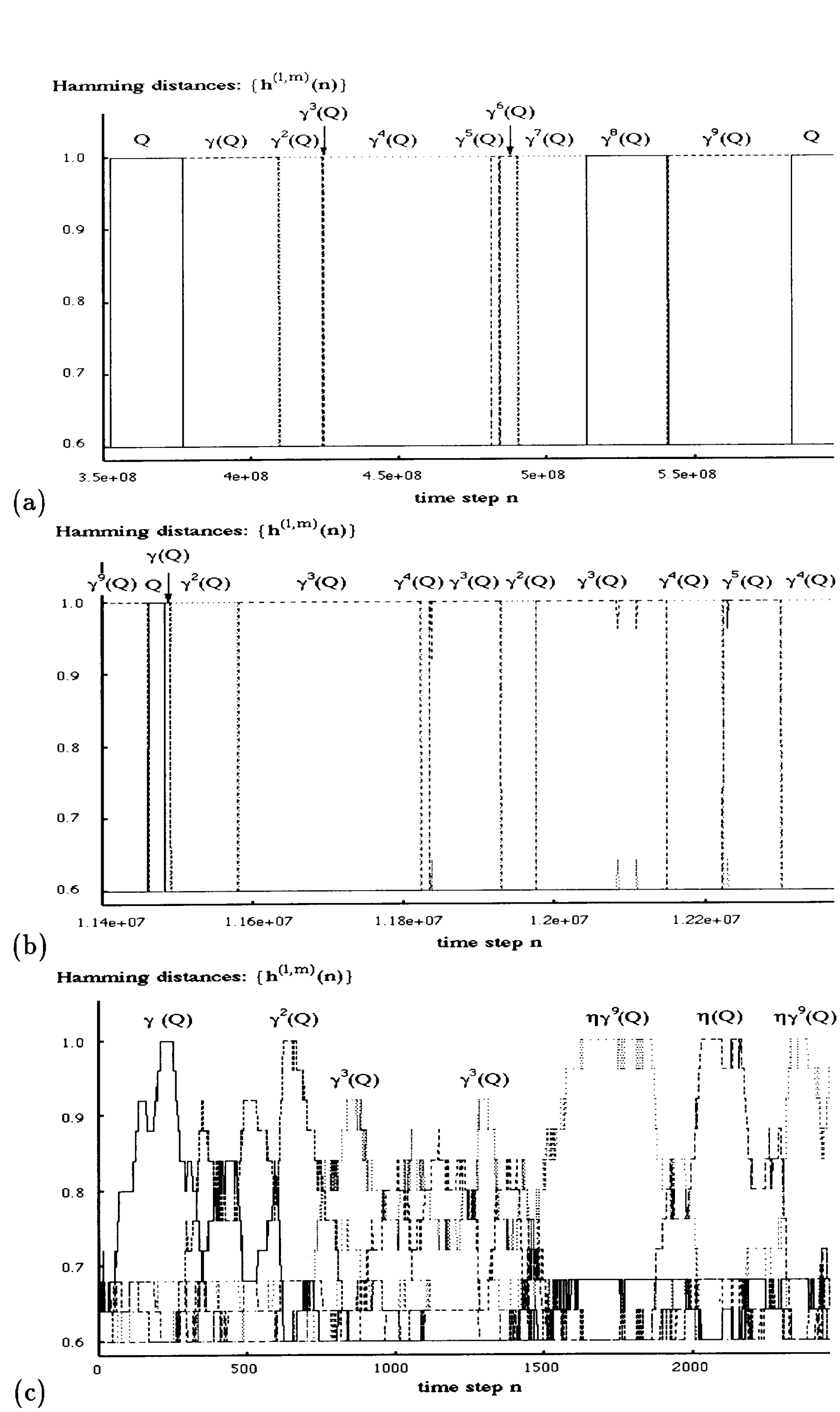


Figure 5.7: Sample sequences $\{J(n)|n = 0, 1, ..\}$ of the temporal network dynamics with (a): $r = 0.730$, (b): $r = 0.702$, and (c): $r = 0.666$ are shown. The averaging duration of eq. (5.12) is set to $w = 500$. The temporal network state is displayed in terms of a set of overlaps between $J(n)$ and the previous localized chaotic attractors $\{\eta^l \circ \gamma^m(Q) \mid l = 0, 1,\ m = 0, .., 9\}$, where the overlap between $J(n)$ and $\eta^l \circ \gamma^m(Q)$ is defined as $h^{(l,m)}(n) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} (2\, \{\eta^l \circ \gamma^m(Q)\}_{ik} - 1)\, (2\, J_{ik}(n) - 1)$.

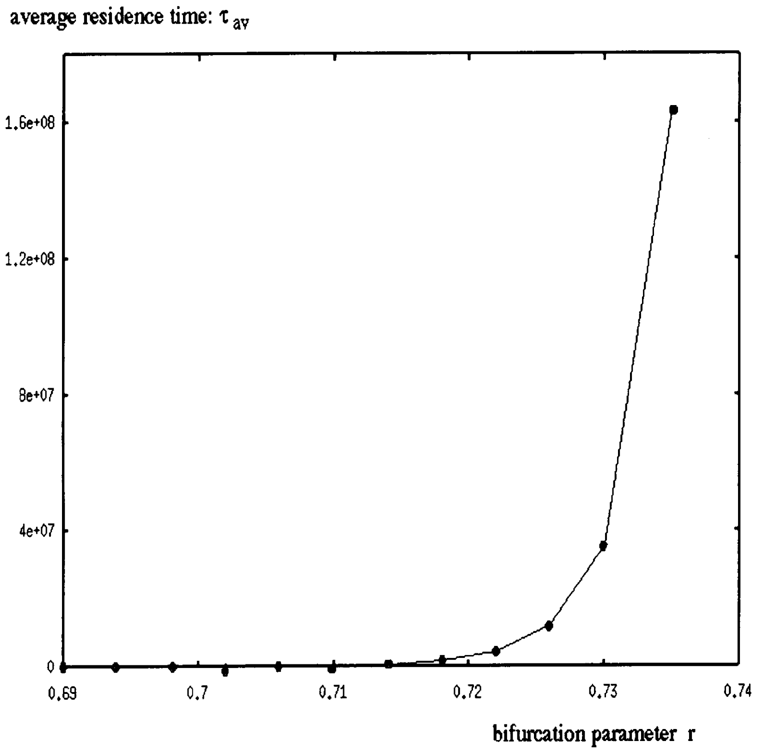


Figure 5.8: The average of residence time τ_{av} in a previous localized chaotic attractor $\{\eta^l \circ \gamma^m(Q) \mid l = 0, 1,\ m = 0, .., 9\}$ is drawn with increasing the value of the bifurcation parameter r.

are observed (see fig. 5.7 (b)). Irregular switches which skip the intermediate states as "$\gamma(Q) \rightleftharpoons \gamma^3(Q)$" or "$\eta \circ \gamma^5(Q) \rightleftharpoons \eta \circ \gamma^8(Q)$" are also observed for systems with small r.

For $r \leq 0.66$, the two conjugate attractors (5.19) and (5.20) finally merge into a fully-connected attractor via second crisis. Chaotic dynamics which "searches" for all the $2N$ previous localized chaotic attractors is observed in this region (see fig. 5.7 (c)).

We remark that, by the first and the second crises, the originally asymmetric attractors $\{\eta^l \circ \gamma^m(O_r) \mid l = 0, 1,\ m = 0, .., N-1\}$ not only increase in size but also increase in symmetry from $\emptyset$-symmetry to G-symmetry. This is a symmetry-increasing bifurcation phenomenon of chaotic attractors which are commonly observed in dynamical systems with symmetry [37, 38, 79].

5.3.5 Remarks on the chaotic itinerancy

In recent studies of high-dimensional dynamical systems, dynamical phenomena called as "chaotic itinerancy" [100, 107, 108, 120, 200, 201, 202, 217] have been extensively studied. The chaotic itinerancy is characterized by high-dimensional chaotic behavior that makes intermittent transitions among a variety of "attractor ruins" which are quasi-stationary sates with effectively low degrees of freedom. Chaotic itinerancy has been discovered in optical systems [100], coupled map models [107, 108] and model neural networks [200, 201, 202]. As is initially suggested by Nozawa [154], the present "chaotic search" for optimization problems can also be considered as the chaotic itinerancy, although the dynamical mechanism such as the structure of the "attractor ruins" has not been well elucidated. Our bifurcation analysis clarifies the dynamical basis of the chaotic itinerancy in the "chaotic search" dynamics in the context of the crisis-induced intermittent switching among the ruins of the previously stable chaotic attractors.

5.4 Application to 5-city TSP

In Section 5.3, we have seen that the crisis-induced switching is the dynamical basis of the "chaotic search" in the 10-city TSP instance. Whereas the previous example shows "chaotic search" among equivalent solutions to the TSP, by using a 5-city TSP instance, this Section shows another example which exhibits "chaotic search" among the optimum solution and the second-optimum solution to TSP.

5.4.1 One-parameter bifurcation of 5-city TSP

As an instance for TSP, 5-city locations of fig. 5.9 are utilized. The 5-city instance was selected among 20 random TSP instances whose 2-dimensional coordinates (x_i, y_i) $(i = 1, 2, .., 5)$ with values between 1/1000 and 1000/1000 were generated by pseudo-random function rand() of the SPARC station 5. The parameter values for the chaotic neural network are set to $(A, B, \omega, \alpha, \beta) = (1.5, 1.0, 0.80, 0.05, 0.018)$.

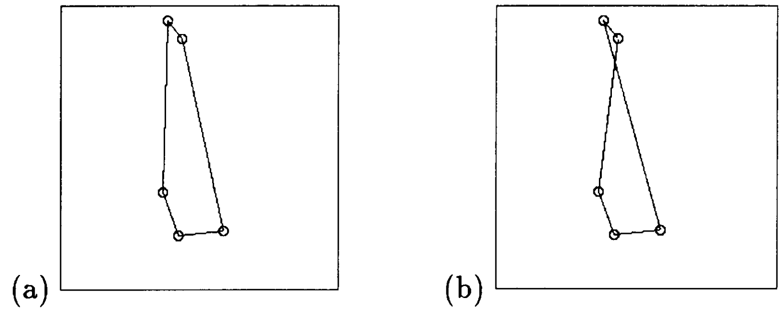


Figure 5.9: Locations of the 5-cities, which are given in 2-dimensional coordinates as (0.3676, 0.3477), (0.4234, 0.1931), (0.5864, 0.2097), (0.3830, 0.9543), (0.4356, 0.8893). (a) represents an optimum tour $Q^{(1)}$ with a total path length of $d = 1.714747$. (b) represents a second-optimum tour $Q^{(2)}$ with a total path length of $d = 1.729555$.

First, for $r = 0.999$, we find the local minimum solutions of the continuous-time Hopfield-Tank neural network by carpet-bombing algorithm. For 100000 samples of random initial conditions which are uniformly distributed over $p(0) \in [0,1]^{N \times N}$, two sets of $2N$ conjugate fixed points $\{\eta^l \circ \gamma^m(q^{(1)}) \mid l = 0, 1, \ m = 0, .., N-1\}$ and $\{\eta^l \circ \gamma^m(q^{(2)}) \mid l = 0, 1, \ m = 0, .., N-1\}$ are found (see Table 5.1). Whereas $q^{(1)}$ is coded as the optimum solution $Q^{(1)}$, $q^{(2)}$ is coded as the second-optimum solution $Q^{(2)}$.

By decreasing the value of the bifurcation parameter r from 0.91 to 0.775, two bifurcation diagrams are drawn respectively from $q^{(1)}$ and $q^{(2)}$ by observing a single neuron state in figs. 5.10 (a) and (c). The figures show essentially similar bifurcation phenomena to the 10-city instance. Namely, in the first stage of the bifurcation diagram, each local minimum $q^{(i)}$ $(i = 1, 2)$ bifurcates into chaotic attractor through period-doubling bifurcation route to chaos. With a further decrease in the bifurcation parameter, each chaotic attractor increases in size. Whereas the repeated occurrence of saddle-node bifurcations gives rise to pairs of stable and unstable periodic solutions generating periodic windows in the bifurcation diagram, the stable periodic attractors also bifurcate into chaotic attractors which soon merge with the original chaotic attractor.

Let us denote the successively observed attractors in each bifurcation diagram as $O_r^{(i)}$ $(i = 1, 2)$. Their bifurcation phenomena are peculiar at the following points.

(i) In each bifurcation diagram, $O_r^{(i)}$ $(i = 1, 2)$ are continually coded as $Q^{(i)}$ (see figs. 5.10 (b) and (d)).

(ii) The basins of the two sets of conjugate attractors $\{\eta^l \circ \gamma^m(O_r^{(i)}) \mid i = 1, 2, \ l = 0, 1, \ m = 0, .., N-1\}$ almost always occupy the entire state space (see Table 5.1).

In other words, bifurcation phenomena which give birth to attractors corresponding to TSP solutions other than $Q^{(1)}$ and $Q^{(2)}$ are rarely observed.

Table 5.1: For 100000 samples of random initial conditions which are uniformly distributed over $p(0) \in [0, 1]^{5\times5}$, basin distribution rates to the two sets of conjugate attractors $\{\eta^l \circ \gamma^m(O_r^{(i)}) \mid i = 1, 2, \ l = 0, 1, \ m = 0, .., 4\}$ are calculated for systems with $r = 0.999$, $r = 0.900$, $r = 0.875$, $r = 0.850$, and $r = 0.825$. The basic attractors $O_r^{(1)}$ and $O_r^{(2)}$ are those observed in the bifurcation diagrams of figs. 5.10 (a) and (c), respectively.

bifurcation parameter	*total basin volumes of* $\{\eta^l \circ \gamma^m(O_r^{(1)}) \mid l = 0, 1, m = 0, .., 4\}$	*total basin volumes of* $\{\eta^l \circ \gamma^m(O_r^{(2)}) \mid l = 0, 1, m = 0, .., 4\}$
r=0.999	71.538 [%]	28.462 [%]
r=0.900	72.966 [%]	27.034 [%]
r=0.875	75.857 [%]	24.143 [%]
r=0.850	75.304 [%]	24.696 [%]
r=0.825	58.758 [%]	41.242 [%]

As the bifurcation parameter is further decreased, the chaotic attractors $O_r^{(i)}$ $(i = 1, 2)$ enlarge their sizes. At the parameter value of $r \cong 0.805$, first, $O_r^{(2)}$ disappears from the

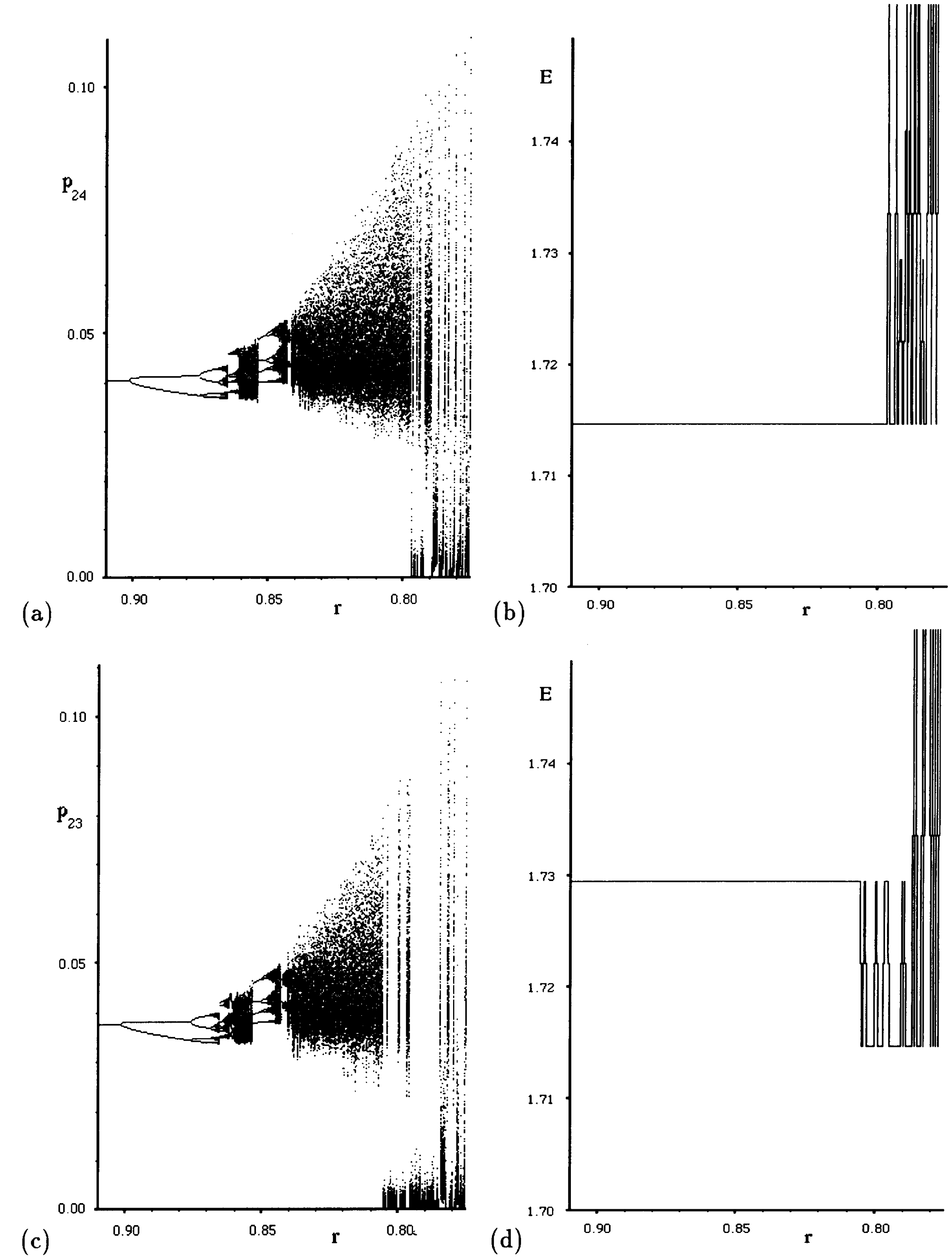


Figure 5.10: A one-parameter bifurcation diagram for the 5-city TSP. The bifurcation parameter value r is decreased 0.91 to 0.775. (a) A one-parameter bifurcation diagram drawn from a local minimum $q^{(1)}$. (b) The cost function values $E(\bar{J}(O_r^{(1)}))$ defined by eq. (5.1) for the successively observed attractors $O_r^{(1)}$ in (a). (c) A one-parameter bifurcation diagram drawn from a local minimum $q^{(2)}$. (d) The cost function values $E(\bar{J}(O_r^{(2)}))$ defined by eq. (5.1) for the successively observed attractors $O_r^{(2)}$ in (c).

bifurcation diagram via a boundary crisis to $O_r^{(1)}$. Then $O_r^{(1)}$ collides with the ruin of the previous chaotic attractor $O_r^{(2)}$ via an interior crisis. The merger of $O_r^{(1)}$ and $O_r^{(2)}$ gives rise to a single attractor $O_r^{(1,2)}$ which exhibits intermittent switching among $Q^{(1)}$ and $Q^{(2)}$ (see fig. 5.12 (a)). Finally, for the parameter region of $r < 0.805$, the conjugate attractors $\{\eta^l \circ \gamma^m(O_r^{(1,2)}) \mid l = 0, 1,\ m = 0, .., N-1\}$ merge into a single attractor via symmetry-increasing crises. The intermittent switches, or "chaotic search", for all the previous localized chaotic attractors $\{\eta^l \circ \gamma^m(Q^{(i)}) \mid i = 1, 2,\ l = 0, 1,\ m = 0, .., N-1\}$ are observed here (see fig. 5.12 (b)). The detailed merging processes are also read from fig. 5.11, which shows the average of switch duration $\{\tau_{1\to 2}, \tau_{2\to 1}, \tau_{SI}\}$ drawn with increasing the value of the bifurcation parameter r, where $\tau_{1\to 2}$ denotes a switch duration from $Q^{(1)}$ to $Q^{(2)}$, $\tau_{2\to 1}$ denotes a switch duration from $Q^{(2)}$ to $Q^{(1)}$, and τ_{SI} denotes a switch duration from $\eta^k \circ \gamma^l(Q^{(i)})$ to $\eta^m \circ \gamma^n(Q^{(j)})$ $((k, l) \neq (m, n))$.

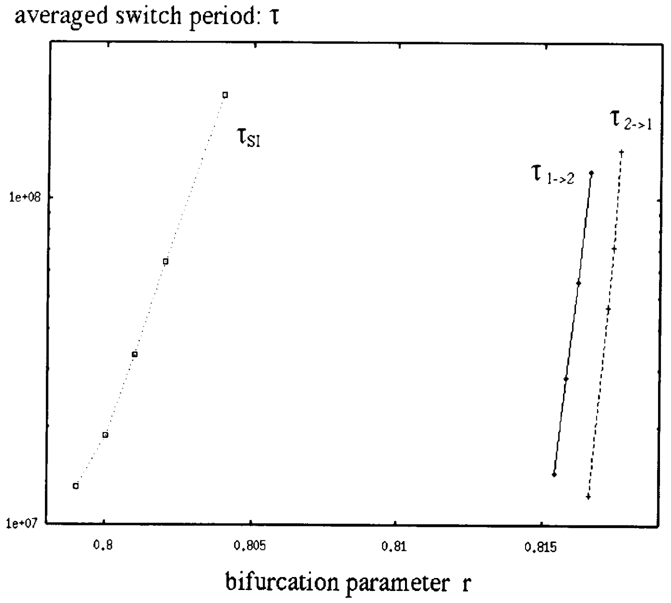


Figure 5.11: The averages of switch durations $\{\tau_{1\to 2}, \tau_{2\to 1}, \tau_{SI}\}$ are drawn with increasing the value of the bifurcation parameter r, where $\tau_{1\to 2}$ denotes the switch duration from $Q^{(1)}$ to $Q^{(2)}$, $\tau_{2\to 1}$ denotes that from $Q^{(2)}$ to $Q^{(1)}$, and τ_{SI} denotes that from $\eta^k \circ \gamma^l(Q^{(i)})$ to $\eta^m \circ \gamma^n(Q^{(j)})$ $((k,l) \neq (m,n))$. With a decrease in the bifurcation parameter, first, switch from $Q^{(2)}$ to $Q^{(1)}$ is observed at $r \simeq 0.8176$. Then switch from $Q^{(1)}$ to $Q^{(2)}$ is recognized at $r \simeq 0.8166$. Finally, switches among the conjugate attractors $\{\eta^l \circ \gamma^m(Q^{(i)}) \mid i = 1, 2,\ l = 0, 1,\ m = 0, .., 4\}$ are observed at $r \simeq 0.804$.

5.4.2 Efficiency of the chaotic search

Let us discuss the efficiency of the chaotic search for the two TSP instances studied in Sections 5.3 and 5.4. In case of the 10-city instance, intermittent switch occurs only among a set of equivalent TSP solutions and hence efficiency of the "chaotic search" for a better

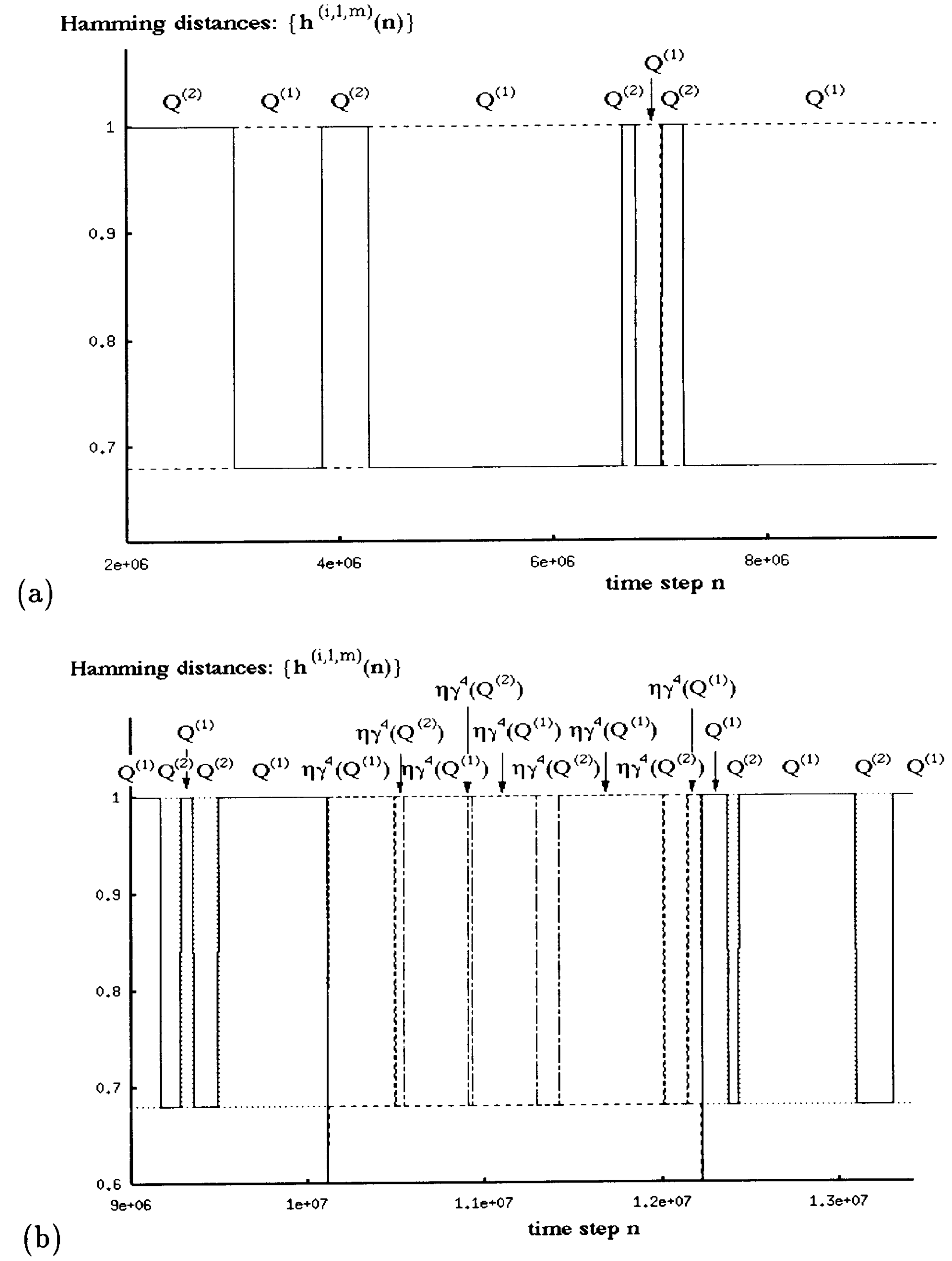


Figure 5.12: Sample sequences $\{J(n)|n = 0, 1, ..\}$ of the temporal network dynamics with (a) $r = 0.806$ and (b) $r = 0.798$ are shown. The averaging duration of eq. (5.12) is set to $w = 300$. The temporal network state is displayed in terms of a set of overlaps between $J(n)$ and the previous localized chaotic attractors $\{\eta^l \circ \gamma^m(Q^{(i)}) \mid i = 1, 2,\ l = 0, 1,\ m = 0, .., 4\}$, where the overlap between $J(n)$ and $\eta^l \circ \gamma^m(Q^{(i)})$ is defined as $h^{(i,l,m)}(n) = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} (2\, \{\eta^l \circ \gamma^m(Q^{(i)})\}_{jk} - 1)\, (2\, J_{jk}(n) - 1)$.

solution can not be discussed. For the 5-city instance, chaotic search occurs among the optimum solution $Q^{(1)}$ and the second-optimum solution $Q^{(2)}$. In this case, efficiency of the chaotic search can be discussed in terms of a transition probability between $Q^{(1)}$ and $Q^{(2)}$. The transition probability $T(Q^{(i)} \rightarrow Q^{(j)})$ [106] represents a probability of transition from $Q^{(i)}$ to $Q^{(j)}$ when the system is trapped in solution $Q^{(i)}$. If the transition probability from $Q^{(2)}$ to $Q^{(1)}$ is higher than that from $Q^{(1)}$ to $Q^{(2)}$, then transitions to optimum solution $Q^{(1)}$ occur more frequently than ones to second-optimum solution $Q^{(1)}$ and hence the dynamics can be considered to be efficient to search for the optimum solution. For $r = 0.80$, the ratio of the transition probabilities $T(Q^{(2)} \rightarrow Q^{(1)})$ and $T(Q^{(1)} \rightarrow Q^{(2)})$ is calculated as $T(Q^{(2)} \rightarrow Q^{(1)})/T(Q^{(1)} \rightarrow Q^{(2)}) = 3.31$, which indicates the higher transition probability to the optimum solution $Q^{(1)}$. Therefore we have confirmed efficiency of the chaotic search for the optimum solution in the 5-city instance, although the problem size is extremely small.

5.5 Conclusions and discussions

5.5.1 Bifurcation scenario

On the basis of the several numerical studies, a simple bifurcation scenario is obtained for chaotic neural networks applied to solve TSP (see also figs. 5.13).

First, a one-parameter family of dynamical systems leading the continuous-time Hopfield-Tank neural network to the chaotic neural network is formulated. With a decrease in the bifurcation parameter, local minimum solutions of the Hopfield-Tank neural network bifurcate into chaotic attractors through period-doubling bifurcation route to chaos. The chaotic attractors are locally distributed in the state space. As the bifurcation parameter is further decreased, the localized chaotic attractors increase in size and eventually merge into a single global attractor via crises. The merging process gives rise to intermittent switching among the previous localized chaotic attractors. Since the previous localized chaotic attractors are interpreted as possible TSP solutions, our bifurcation studies find that the crisis-induced intermittent switches underly the "chaotic search" for TSP solutions.

We remark that the present bifurcation scenario is obtained only from two instances of small-scale TSP and may not necessarily provide a general bifurcation theory of the chaotic neural networks for the optimization problem. We believe, however, that our bifurcation scenario is still valid for describing the essential features of the chaotic neurodynamics which may efficiently work for a wide class of TSPs, since we have confirmed essentially similar bifurcation phenomena in several other instances of TSP.

5.5.2 Optimization capability

On the basis of our bifurcation scenario, let us discuss the optimization capability of the "chaotic search." The observed bifurcation phenomena are peculiar in the sense that they rarely give birth to attractors corresponding to TSP solutions except those observed in the local minimum solutions of the underlying Hopfield-Tank neural network. It is most probable that the previous localized chaotic attractors in the "chaotic search" region are born from period-doubling bifurcations of the local minimum solutions of the Hopfield-Tank neural network and hence they have almost one-to-one correspondence with the local minima. This bifurcation property can be also observed in other instances including higher-dimensional TSPs, although theoretical understanding of this property is an important open question. In the sense that the chaotic dynamics seeks for a better TSP solution among local minimum solutions of the Hopfield-Tank neural network without being trapped in one solution, our bifurcation scenario proves the "chaotic search" capability which overcomes the weakness of the Hopfield-Tank search.

Our bifurcation studies also imply that the efficiency of the "chaotic search" strongly depends upon the complex linkage structure of a variety of previous localized chaotic attractors. A large number of dynamical paths leading to a global minimum may produce efficient search for the global minimum, whereas a large number of paths leading to local minima may provide poor results. From this viewpoint, we have studied the "chaotic search" capability for optimum solution by calculating the transition probabilities among the switching solutions. For the 10-city instance, "chaotic search" occurs only among a set of equivalent global minimum solutions and hence efficiency of the chaotic switch from local minima to global minimum can not be discussed. For the 5-city instance, chaotic search occurs among the optimum solution and the second-optimum solution. The transition probability analysis clarified that the chaotic switch to the optimum solution occurs more frequently than that to the second-optimum solution. We therefore confirmed the efficiency of the chaotic search for global minimum solution in this toy problem with 5 cities.

We remark that the present study on the efficiency of the "chaotic search" provides only preliminary results because our experiments deal with only two small-scale instances of TSP. It is an important future problem to examine the efficiency of the chaotic search for a variety of TSP instances including large-scale problems with further statistical studies based on the calculation of the transition probabilities or other statistical quantities.

It is also necessary to compare the chaotic search algorithm with other conventional optimization algorithms such as the heuristic algorithms [124, 127], the genetic algorithms [73], and so forth and clarify disadvantages as well as advantages of the chaotic search method.

5.5.3 Parameter tuning

Another problem to be settled for practical application of the present method is the difficulty of choosing good parameter values for $(A, B, \alpha, \omega, r)$ that give rise to efficient "chaotic search." In this study, we have adopted the parameter values of Nozawa [154]. Whereas there is no systematic way for determining good parameter values for (A, B, α, ω), the present bifurcation analysis at least provides a hint for tuning the bifurcation parameter value r. Namely, by following the bifurcation procedure of one of the local minima of the Hopfield-Tank neural network, one may find the bifurcation parameter region where the chaotic attractors merge with others via a series of crises and an efficient "chaotic search" takes place. This reduces the amount of labor for adjusting the bifurcation parameter value r by trial and error methods.

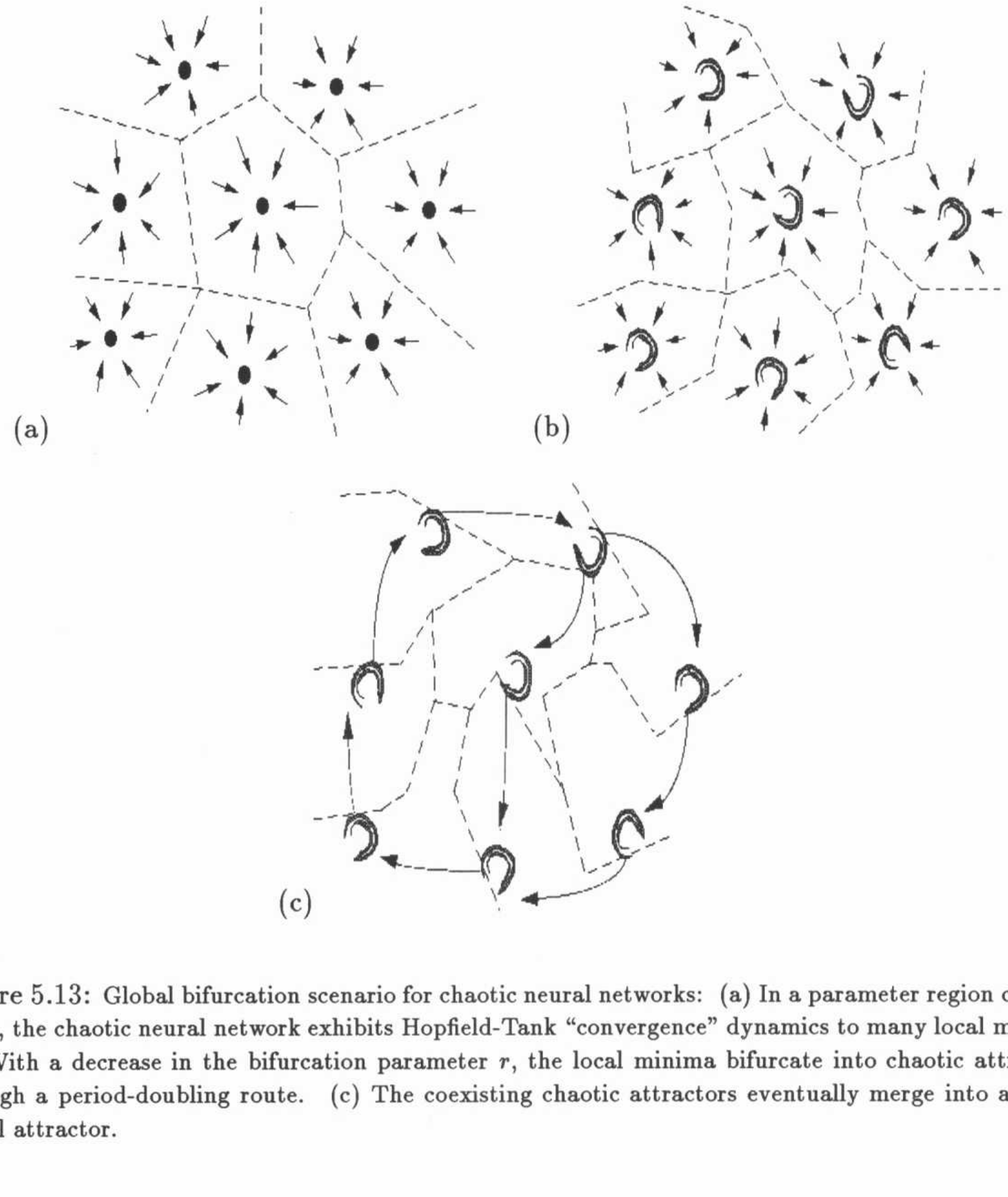


Figure 5.13: Global bifurcation scenario for chaotic neural networks: (a) In a parameter region close to $r = 1$, the chaotic neural network exhibits Hopfield-Tank "convergence" dynamics to many local minima. (b) With a decrease in the bifurcation parameter r, the local minima bifurcate into chaotic attractors through a period-doubling route. (c) The coexisting chaotic attractors eventually merge into a single global attractor.

Chapter 6

Simulated annealing in chaotic neural network

abstract: Chaotic simulated annealing algorithm for combinatorial optimization problems is examined in the light of the global bifurcation structure of the chaotic neural networks. We show that the result of the chaotic simulated annealing algorithm is primarily dependent upon the global bifurcation structure of the chaotic neural networks and unlike the stochastic simulated annealing *infinitely slow* chaotic annealing does not necessarily provide an optimum result. As the improved algorithms, adaptive chaotic simulated annealing algorithm and learning algorithm are introduced to chaotic neural networks. Using several instances of traveling salesman problems, efficiency of the modified algorithms is demonstrated.

6.1 Introduction

Due to the recent successful results in a variety of engineering problems, application of chaotic neural networks [12] to combinatorial optimization problems has received a great deal of attention [35, 36, 83, 103, 154]. As is described in detail in Chapter 5, the idea of optimization by chaotic neural networks can be briefly summarized as follows. In the continuous state space of the chaotic neural network, every possible solution of an optimization problem is embedded. By following chaotic wondering orbit which visits a variety of the solutions, chaotic dynamics continually searches for the optimum or near-optimum solution. In contrast with the conventional Hopfield-Tank neural network search [94], the non-equilibrium chaotic search overcomes the local minimum problem. Compared to the stochastic search system [71, 114] whose search space is essentially the same with the whole state space, the chaotic search dynamics is confined in a relatively low-dimensional fractal space, which seems to realize an efficient search for a variety of optimization problems such as the traveling salesman problem (TSP) [35, 36, 83, 103, 154].

For the chaotic neural network approach to optimization problems, it is natural to introduce the idea of *simulated annealing* [114], which is the physical model for gradually cooling the thermodynamical system to its ground state. Technically, it is quite important to gradually cool down the chaotic dynamics to a possibly optimum state by simulated annealing, since the chaotic search is basically everlasting.

Towards the simulated annealing in chaotic neural network, chaotic simulated annealing (CSA) algorithm has been recently developed by Chen & Aihara [35]. In the CSA algorithm, the chaotic dynamics is harnessed by a cooling algorithm of a bifurcation parameter. Gradual cooling of the bifurcation parameter controls the chaotic search dynamics to converge to a stable equilibrium state with a possibly optimum or near-optimum solution. The experimental studies in [35] demonstrate the efficiency of the CSA algorithm which obtains fairly good solutions of TSP.

The aim of the present Chapter is to re-examine the efficiency of the CSA algorithm in the light of the global bifurcation structure of the chaotic neural networks studied in Chapter 5. On the basis of the bifurcation analyses, we argue that the result of the chaotic annealing is primarily dependent upon the global bifurcation structure of the chaotic neural networks and show that *infinitely slow* chaotic annealing does not necessarily provide an optimum result. This is unlike the optimization property of the stochastic simulated annealing which realizes a global optimization in the limit of *infinitely slow* annealing [71]. As the improved algorithms for CSA, adaptive chaotic simulated annealing algorithm and learning algorithm are introduced to chaotic neural network. Using several instances of TSPs, efficiency of the modified algorithms is demonstrated.

6.2 Chaotic neural network for TSP

Let us consider an N-city symmetric TSP [124]: "Given an $N \times N$ symmetric matrix (d_{ij}) of distances between a set of N cities $(i, j = 1, 2, .., N)$, find a minimum-length tour that visits each city exactly once."

As is described in detail in Chapter 5, chaotic neural network that solves the TSP is described in terms of an $N \times N$-dimensional mapping:

$$p_{ik}(n+1) = r\, p_{ik}(n) + (1-r)\, \sigma\Big(\sum_{j=1}^{N}\sum_{l=1}^{N} T_{ik,jl}\, p_{jl}(n) + I_{ik}\Big), \tag{6.1}$$

where p_{ik} stands for an internal state of the (i, k)-neuron $(i, k = 1, .., N)$, r $(0 < r < 1)$ stands for a decay parameter, and $\sigma(x) = 0.5 + 0.5\tanh(x/\epsilon)$. The synaptic connections $T_{ik,jl}$ are given by

$$\begin{aligned} T_{ik,jl} &= -A(\delta_{ij}(1-\delta_{kl}) + \delta_{kl}(1-\delta_{ij})) - Bd_{ij}(\delta_{l\,k+1} + \delta_{l\,k-1}), && (6.2)\\ T_{ik,ik} &= -2\omega A, && (6.3)\\ I_{ik} &= 2\alpha A, && (6.4)\end{aligned}$$

where A and B are balancing parameters of the constraint term and the tour-length term of the TSP cost function, α is a control parameter for excitation level of neurons, and ω is a negative self-feedback parameter.

By wandering around a variety of temporal network firing states $\{\ \rho_{ik}(n) = (1/w) \sum_{j=0}^{w-1} p_{ik}(n-j) \mid i,k = 1,..,N\ \}$ (w : averaging duration), which are coded into possible TSP solutions $J(n) = \{\ J_{ik}(n) \mid i,k = 1,..,N\}$ by $J_{ik}(n) = 1[\rho_{ik}(n) - \rho^*]$ ($1[x] = 1$ if $x \geq 0$, $1[x] = 0$ if $x < 0$, ρ^* : the N-th largest among $\{\rho_{ik}(n)\}$), the chaotic neural network searches for the optimum solution among a variety of TSP solutions.

In Chapter 5, we have obtained the following global bifurcation scenario for chaotic neural networks that solve TSP: We take the decay parameter r as the bifurcation parameter. First, there is a bifurcation parameter region with $r \approx 1$, where nonlinear dynamics of the chaotic neural network of eq. (6.1) becomes similar to that of the continuous-time Hopfield-Tank neural network which exhibits "convergence" dynamics to the local minimum solutions. With a decrease in the bifurcation parameter r, the local minimum solutions bifurcate into chaotic attractors through period-doubling bifurcation route to chaos. The chaotic attractors are initially localized in the state space and eventually merge into a single global attractor via a series of crises [80]. The merging process gives rise to intermittent switching among the previous localized chaotic attractors and the global "chaotic search" for various TSP solutions takes place.

6.3 Chaotic simulated annealing

6.3.1 Slow annealing

In the CSA algorithm [35], in order to terminate the chaotic search procedure and to obtain the final solution, the chaotic dynamics is eventually controlled to converge to a stable equilibrium state by a gradual cooling of the bifurcation parameter r. On the basis of our bifurcation scenario, the CSA algorithm for TSP can be formulated as follows:

[Slow CSA algorithm]

STEP 1 (*Initial condition*):
Choose random initial condition $p(0) \in [0,1]^{N\times N}$ and set $n = 0$ and $r(0) = r_s$, where r_s stands for a bifurcation parameter value that gives rise to "chaotic search" dynamics.

STEP 2 (*Update of chaotic neural network*):

$$\begin{aligned} p_{ik}(n+1) &= r(n)\, p_{ik}(n) + (1-r(n))\, \sigma\Big(\sum_{j=1}^{N}\sum_{l=1}^{N} T_{ik,jl}\, p_{jl}(n) + I_{ik}\Big), & (6.5)\\ r(n+1) &= (1-\beta)(r(n) - r_g) + r_g. & (6.6) \end{aligned}$$

β ($0 < \beta < 1$) stands for an annealing speed parameter and r_g stands for a bifurcation parameter value that gives rise to Hopfield-Tank "convergence" dynamics.

STEP 3 (*Termination*):
If $|r - r_g| < 0.01$, stop. Else, set $n = n + 1$ and go to STEP 2.

With a random initial condition $p(0)$ and with a bifurcation parameter $r(0) = r_s$, at the first stage of the annealing, the network searches for TSP solutions by chaotic wandering dynamics. As the annealing proceeds with $r(n) \to r_g$, the chaotic search eventually converges to a single equilibrium solution.

Fig. 6.2 shows the result of the CSA algorithm applied to 5-city TSP of fig. 6.1. The 5-city instance was selected among 100 random TSP instances whose 2-dimensional coordinates $(x_i,\ y_i)$ $(i = 1, 2, .., 5)$ with values between 1/1000 and 1000/1000 were generated by pseudo-random function **rand()** of the SPARC station 5.

As the annealing speed β is decreased, we see that the convergence rate to the optimum solution is decreased. As is explained in detail in the next subsection, this phenomenon is due to the global bifurcation structure of the chaotic neural network.

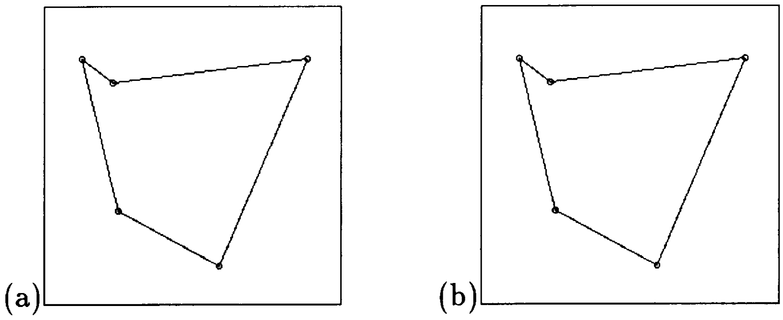


Figure 6.1: Locations of the 5-cities, which are given in 2-dimensional coordinates as (0.1768, 0.2233), (0.9348, 0.6305), (0.1561, 0.5661), (0.5793, 0.0830), (0.0358, 0.6269). (a) represents an optimum tour $Q^{(1)}$ with a total path length of $d = 2.422713$, while (b) represents a second-optimum tour $Q^{(2)}$ with a total path length of $d = 2.456266$.

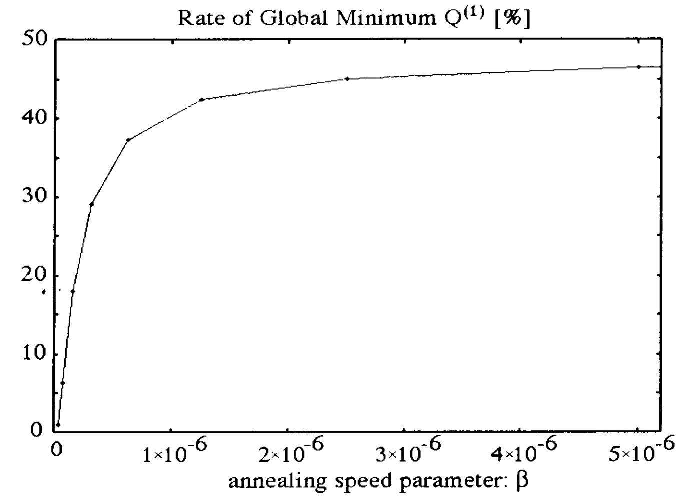


Figure 6.2: The global optimization rate of the CSA algorithm for the 5-city TSP is drawn with decreasing the annealing speed parameter β. The system parameters are set to $(A,\ B,\ \omega,\ \alpha,\ \epsilon,\ r_g,\ r_s) =$ (1.5, 1.0, 0.70, 0.07, 0.018, 0.95, 0.75).

6.3.2 Hierarchical merging structure of TSP solutions

Let us study the global bifurcation structure of the 5-city TSP instance. For simplicity of our discussion, symmetry of the chaotic neural network is not considered in this Chapter.

First, for $r = 0.999$ that gives rise to the Hopfield-Tank convergence dynamics, two local minimum solutions, the optimum solution $Q^{(1)}$ and the second-optimum solution $Q^{(2)}$, are found by the carpet-bombing algorithm. With a decrease in the bifurcation parameter r, the two solutions, $Q^{(1)}$ and $Q^{(2)}$, bifurcate into chaotic attractors through period-doubling bifurcation route to chaos. As is shown in Table 6.1, in this bifurcation process, the two solutions seem to be the only observable attractors of the chaotic neural network. In other words, bifurcation phenomena which give birth to attractors corresponding to TSP solutions other than $Q^{(1)}$ and $Q^{(2)}$ are rarely observed.

The two attractors are initially localized in the state space and eventually merge into a single attractor via crises. In fig. 6.3, the detailed merging processes are clarified by computing the averaged switch duration $\{\tau_{1\to2}, \tau_{2\to1}\}$ with an increasing bifurcation parameter r, where $\tau_{1\to2}$ denotes a switch duration from $Q^{(1)}$ to $Q^{(2)}$ and $\tau_{2\to1}$ denotes a switch duration from $Q^{(2)}$ to $Q^{(1)}$. According to the switch duration curves, first, the optimum solution $Q^{(1)}$ touches the separatrix of $Q^{(1)}$ and $Q^{(2)}$ and looses its stability via a boundary crisis. Then, the second-optimum solution $Q^{(2)}$ merge with the ruin of the optimum solution $Q^{(1)}$ via an interior crisis (see fig. 6.4). The merger of $Q^{(1)}$ and $Q^{(2)}$ gives rise to intermittent switch dynamics among $Q^{(1)}$ and $Q^{(2)}$ (see fig. 6.5).

Table 6.1: For 100000 samples of random initial conditions uniformly distributed over $p(0) \in [0,1]^{5\times 5}$, basin distribution rates to the two solutions, $\{Q^{(1)}$ and $Q^{(2)}\}$, are calculated for systems with $r = 0.999$, $r = 0.950$, $r = 0.900$, $r = 0.875$, and $r = 0.850$.

bifurcation parameter	*total basin volumes of* $Q^{(1)}$	*total basin volumes of* $Q^{(2)}$
r=0.999	51.43 [%]	48.57 [%]
r=0.950	51.19 [%]	48.81 [%]
r=0.900	51.43 [%]	48.57 [%]
r=0.875	50.58 [%]	49.42 [%]
r=0.850	48.98 [%]	51.02 [%]

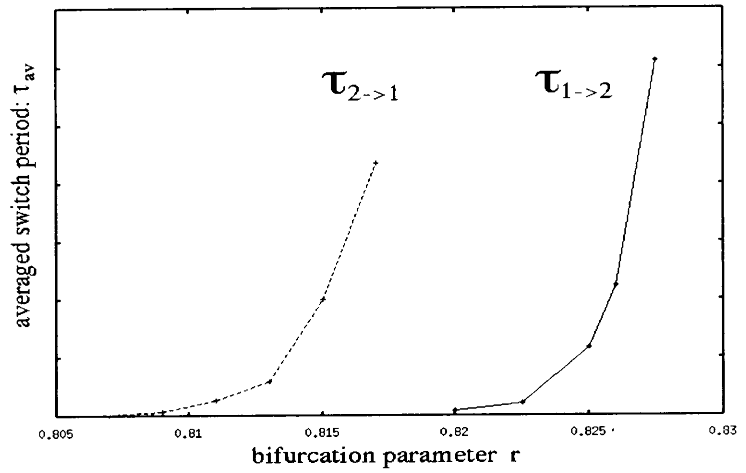


Figure 6.3: The averages of switch durations $\{\tau_{1\to2}, \tau_{2\to1}\}$ are drawn with increasing the value of the bifurcation parameter r, where $\tau_{1\to2}$ denotes a switch duration from $Q^{(1)}$ to $Q^{(2)}$ and $\tau_{2\to1}$ denotes a switch duration from $Q^{(2)}$ to $Q^{(1)}$.

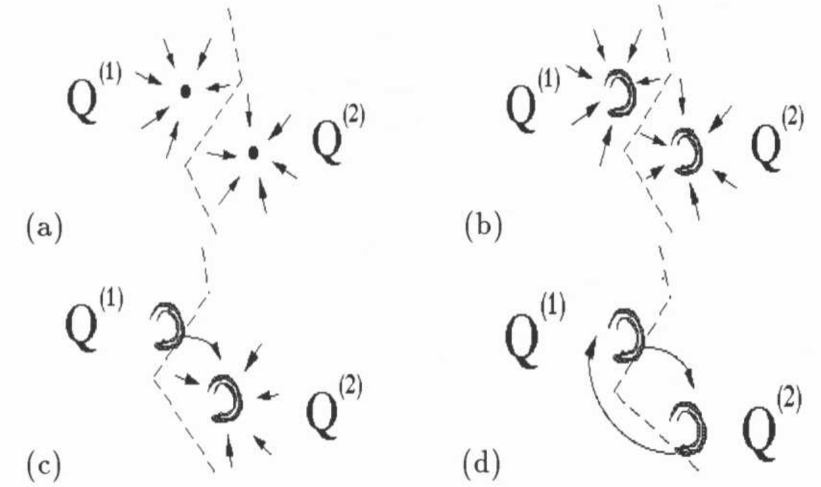


Figure 6.4: Schematic illustration of the merging process of $Q^{(1)}$ and $Q^{(2)}$. (a) First, there exist two local minimum solutions, $Q^{(1)}$ and $Q^{(2)}$, in the Hopfield-Tank dynamics region. (b) With a decrease in the bifurcation parameter r, two local minima bifurcate into chaotic attractors through period doubling bifurcation route to chaos. The two chaotic attractors are initially localized in the state space. (c) First, the optimum solution $Q^{(1)}$ touches the separatrix of $Q^{(1)}$ and $Q^{(2)}$ and looses its stability via a boundary crisis. (d) Then, the second-optimum solution $Q^{(2)}$ merge with the ruin of the optimum solution $Q^{(1)}$ via an interior crisis.

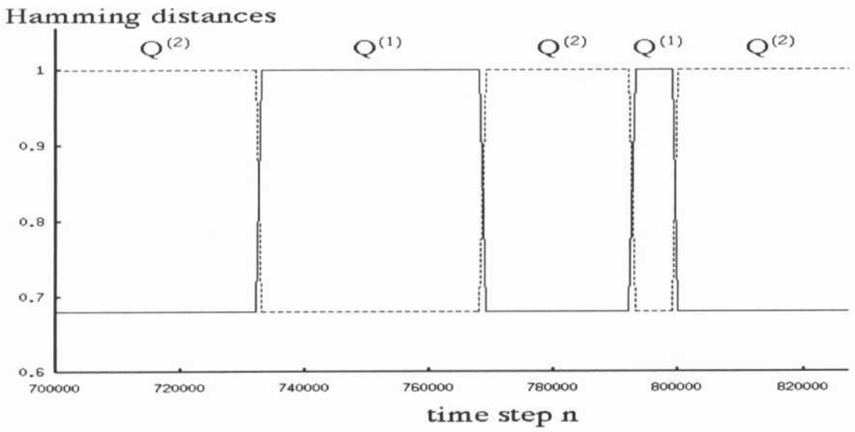


Figure 6.5: Intermittent switch dynamics among $Q^{(1)}$ and $Q^{(2)}$.

The merger of $Q^{(1)}$ and $Q^{(2)}$ can be schematically illustrated in a binary tree structure of fig. 6.6. Notice that, in this merging process, whereas the global minimum solution $Q^{(1)}$ has an unstable parameter region, the second minimum solution $Q^{(2)}$ is continually stable until the final merger. An infinitely slow annealing always provides the second minimum solution $Q^{(2)}$, because in the unstable parameter region of $Q^{(1)}$ every slow annealing is trapped in the second minimum solution $Q^{(2)}$.

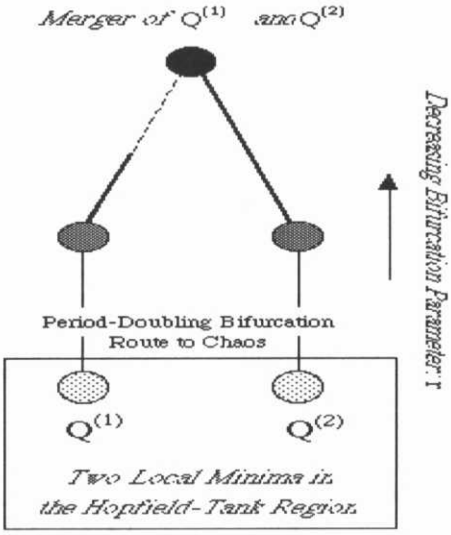

Merger of $Q^{(1)}$ and $Q^{(2)}$
Decreasing Bifurcation Parameter: r
Period-Doubling Bifurcation Route to Chaos
$Q^{(1)}$
$Q^{(2)}$
Two Local Minima in the Hopfield-Tank Region

Figure 6.6: Binary tree representing the merging process of two solutions, $Q^{(1)}$ and $Q^{(2)}$. The continual solid line indicates the branch of the stable solution $Q^{(2)}$, the broken line indicates the branch of the unstable solution $Q^{(1)}$ which lost its stability via a boundary crisis, and the node of the two branches indicates the merger of $Q^{(1)}$ and $Q^{(2)}$.

Let us consider a general case of annealing multiple-attractor systems. As in the previous discussion, hierarchical merging process of multiple chaotic attractors $\{Q^{(1)}, Q^{(2)}, ..\}$ can be schematically represented in a binary tree structure of fig. 6.7. Notice that, in the merging process, there is only a single chaotic attractor which is continually stable until the final merger. An infinitely slow annealing which traces only a stable solution provides such a continual attractor as the final solution. This implies that an *infinitely slow* annealing does not necessarily provide an optimum result, since the optimum solution does not always survive until the final merger. With an analogy from the stochastic simulated annealing [71], it has been conjectured that the CSA algorithm also provides an optimum result by infinitely slow annealing. The present result provides a counter-example for this conjecture.

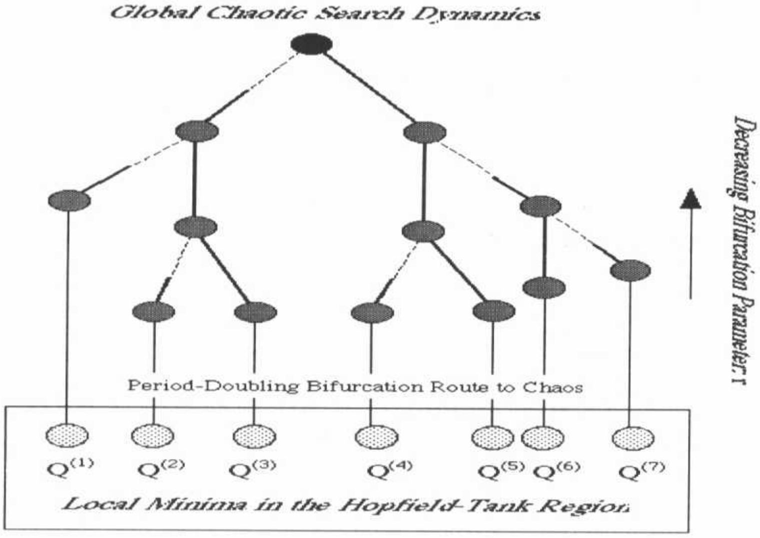

Global Chaotic Search Dynamics
Decreasing Bifurcation Parameter r
Period-Doubling Bifurcation Route to Chaos
Q(1)
Q(2)
Q(3)
Q(4)
Q(5)
Q(6)
Q(7)
Local Minima in the Hopfield-Tank Region

Figure 6.7: Binary tree representing the hierarchical merging structure of multiple TSP solutions $\{Q^{(1)}, Q^{(2)}, ..\}$. The solid line indicates the branch of a stable solution, the broken line indicates the branch of an unstable solution which lost its stability via a boundary crisis, and the node of the two branches indicates the merger of the two solutions.

6.4 Adaptive chaotic annealing

In order to improve the conventional CSA algorithm, we introduce the following adaptive chaotic simulated annealing (adaptive CSA) algorithm:

[Adaptive CSA algorithm]

STEP 1 (*Initial condition*):
Choose random initial condition $p(0) \in [0,1]^{N\times N}$ and set $n = 0$ and $r(0) = r_s$.

STEP 2 (*Update of chaotic neural network*):

$$\begin{aligned} p_{ik}(n+1) &= r(n)\, p_{ik}(n) + (1-r(n))\, \sigma\Big(\sum_{j=1}^{N}\sum_{l=1}^{N} T_{ik,jl}\, p_{jl}(n) + I_{ik}\Big), && (6.7)\\ r(n+1) &= (1-\beta)(r(n)-r_g)+r_g \qquad (\text{ if } E(J(n)) \le E_{th} \text{ }), && (6.8)\\ r(n+1) &= (1-\beta)(r(n)-r_s)+r_s \qquad (\text{ if } E(J(n)) > E_{th} \text{ }). && (6.9)\end{aligned}$$

E_{th} stands for a threshold value and $E(\cdot)$ stands for a cost function defined by using temporal network firing state $J(n)$ as

$$\begin{aligned} E(J(n)) = \frac{A}{2}\sum_{i=1}^{N}\{\sum_{k=1}^{N} J_{ik}(n) - 1\}^2 + \frac{A}{2}\sum_{k=1}^{N}\{\sum_{i=1}^{N} J_{ik}(n) - 1\}^2 + \\ \frac{B}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} d_{ij} J_{ik}(n)\{J_{j\,k+1}(n) + J_{j\,k-1}(n)\}. \qquad (6.10)\end{aligned}$$

STEP 3 (*Termination*):
If $|r - r_g| < 0.01$, stop. Else, set $n = n + 1$ and go to STEP 2.

The adaptive CSA algorithm utilizes "chaotic search" dynamics to seek for a TSP solution that has lower cost than the threshold value E_{th}. When such a solution is found, the algorithm promptly tunes the bifurcation parameter r to the Hopfield-Tank "convergence" region and cools down the network dynamics to the equilibrium state. If the trapping is failed, chaotic search for another satisfactory solution is repeated.

Fig. 6.8 shows an example of the adaptive annealing algorithm applied to the 5-city TSP. The annealing speed is set to $\beta = 0.02$ and the threshold value is set to the optimum tour length $E_{th} = 2.422713$. When "chaotic search" finds the optimum solution $Q^{(1)}$, the adaptive CSA algorithm promptly tunes the bifurcation parameter r to the Hopfield-Tank "convergence" region and successfully stabilize the optimum solution.

Table 6.2 shows results of the adaptive CSA algorithm applied to random 20-city, 30-city, 40-city, 50-city, and 60-city TSP. For each random TSP, 2-dimensional coordinates of the city locations $(x_i,\, y_i)$ with values between 1/1000 and 1000/1000 were generated by pseudo-random function **rand()** of the SPARC station 5. The threshold value is set as $E_{th} = (1 + \varepsilon)\ C_{HK}$ (ε: gap parameter) using the Held-Karp lower bound C_{HK} [87]. The adaptive CSA algorithm provides much better solutions with fewer numbers of computation steps compared with the conventional slow CSA algorithm.

Fig. 6.9 shows dependence of the computational steps S against the problem size N, where the log-log plot clarifies a scaling property of $S \propto N^{2.55}$. Although our present study is based on relatively small-scale TSPs, this scaling property demonstrates the efficiency of the adaptive CSA algorithm that finds an approximately good TSP solution within a polynomial computational time.

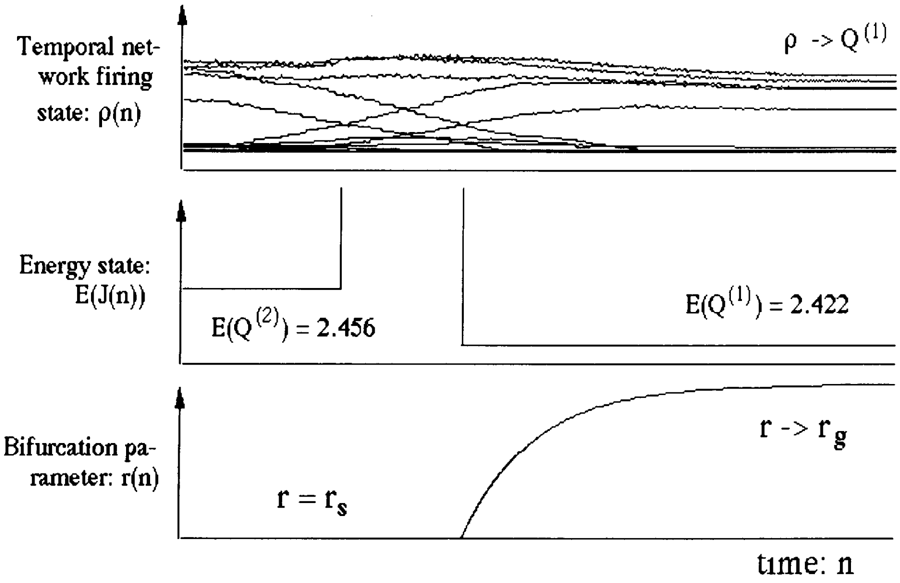


Figure 6.8: Example of the adaptive annealing algorithm applied to 5-city TSP. The annealing speed and the threshold value are set to $\beta = 0.02$, $E_{th} = 2.422713$, and other parameters are set same as fig. 6.2.

Table 6.2: Results of the adaptive CSA algorithm and the slow CSA algorithm against 20 instances of random 30-city, 40-city, 50-city, and 60-city TSP, where 2-dimensional coordinates of the random city locations $(x_i,\ y_i)$ with values between 1/1000 and 1000/1000 were generated by pseudo-random function **rand()** of the SPARC station 5. For each instance, 100 sets of random initial conditions are prepared. The parameters of the adaptive CSA algorithm are fixed to $(A,\ B,\ \omega,\ \alpha,\ \epsilon,\ r_g,\ r_s,\ \beta,\ \varepsilon,\ w) = (1.0,\ 1.0,\ 0.75,\ 0.1,\ 0.018,\ 0.95,\ 0.7,\ 0.04,\ 0.1,\ 100)$. The annealing speed of the slow CSA algorithm is set to $\beta = 0.0005$ for 30- and 40-city TSP, $\beta = 0.0002$ for 50-city TSP, $\beta = 0.0001$ for 60-city TSP. The averaged tour length obtained by the adaptive CSA algorithm is denoted by its ratio to the averaged tour length obtained by the slow CSA algorithm.

	Adaptive CSA	*Slow CSA*
averaged tour length for random 30-city TSP	0.874	1.00
averaged computation steps	7254.2	11041.0
averaged tour length for random 40-city TSP	0.849	1.00
averaged computation steps	9506.9	11041.0
averaged tour length for random 50-city TSP	0.860	1.00
averaged computation steps	19398.2	27605.0
averaged tour length for random 60-city TSP	0.849	1.00
averaged computation steps	30338.0	55212.0

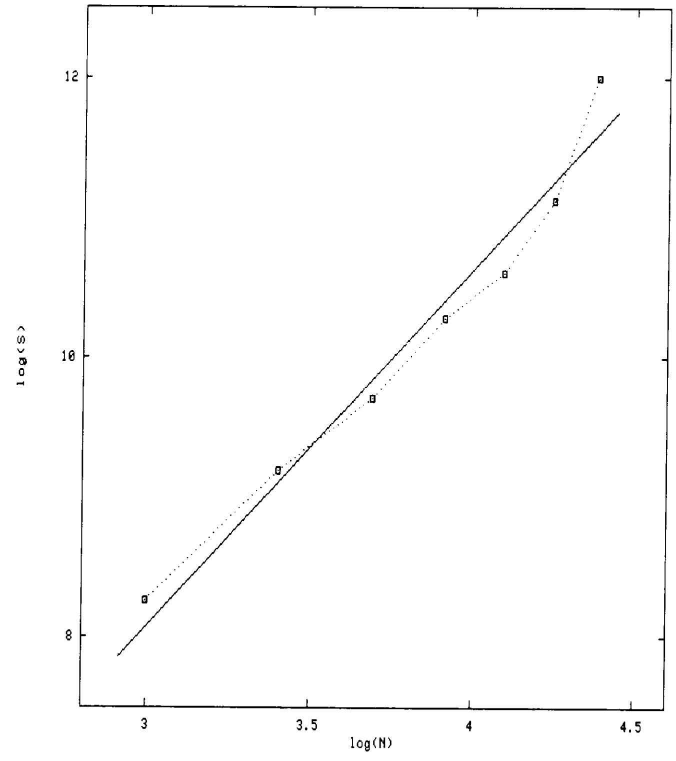


Figure 6.9: Dependence of the computational step S against the problem size N. The log-log plot is well approximated by the least-square-error line of $\log(S) = 2.55 \log(N - 3.82) + 10.17$.

6.5 Learning algorithm

In Section 6.4, we have introduced the adaptive CSA algorithm as an improved algorithm for CSA. Although the adaptive algorithm was shown to be effective for finding near-optimum solutions, there is no guarrentee for finding the global minimum. We also note that the conventional CSA algorithm and the adaptive CSA algorithm are based on the controlling algorithm of a single bifurcation parameter. It is a challenging but worthwhile investigation to develop an algorithm which directly controls the asymptotic measure of the chaotic neural networks. Controlling the asymptotic measure of the chaotic search dynamics to eventually converge to the optimum state may provide us with more natural annealing algorithm. Towards this annealing, we introduce the following learning algorithm for chaotic neural network:

[Learning algorithm for chaotic neural network]

STEP 1 (*Initial condition*):
Choose random initail condition $p(0) \in [0,1]^{N\times N}$ and set $n = 0$ and $E_{min} = E(J(0))$.

STEP 2 (*Update of chaotic neural network*):

$$p_{ik}(n+1) = rp_{ik}(n) + (1-r)\sigma(\sum_{j=0}^{N-1}\sum_{l=0}^{N-1} T_{ik,jl}p_{jl}(n) + I_{ik})$$

STEP 3 (*Learning*):
If $E(J(n+1)) \leq (1+\mu)E_{min}$;

$T_{ik,jl} = T_{ik,jl} + \gamma$ (for $p_{ik}, p_{jl} \in$ firing or $p_{ik}, p_{jl} \notin$ firing),
$T_{ik,jl} = T_{ik,jl} - \gamma$ (for $p_{ik} \in$ firing, $p_{jl} \notin$ firing or $p_{ik} \notin$ firing, $p_{jl} \in$ firing).

Else;

$T_{ik,jl} = T_{ik,jl} - \theta$ (for $p_{ik}, p_{jl} \in$ firing or $p_{ik}, p_{jl} \notin$ firing)
$T_{ik,jl} = T_{ik,jl} + \theta$ (for $p_{ik} \in$ firing, $p_{jl} \notin$ firing or $p_{ik} \notin$ firing, $p_{jl} \in$ firing).

μ, γ, and θ ($\mu, \gamma, \theta > 0$) stand for learning parameters.

STEP 4 (*Termination*):
If $J(n+1) = J(n) = \cdots = J(n-99)$, stop. Else, set $E_{min} = \min(E_{min}, E(J(n+1)))$ and $n = n+1$ and go to STEP 2.

The algorithm is based on the learning rule introduced by Watanabe *et al.* for the memory acquisition in chaotic neural network [207]. As is illustrated in fig. 6.10, the learning algorithm has the following functions:

1. For a solution $J(n)$ which has lower cost than the minimum value $(1+\mu)E_{min}$, stabilize the solution by strengthening the associated synaptic connections (**local search effect**).

2. For a bad solution $J(n)$ which has higher cost than $(1+\mu)E_{min}$, destabilize the solution by weakening the associated synaptic connections (**tabu search effect**).

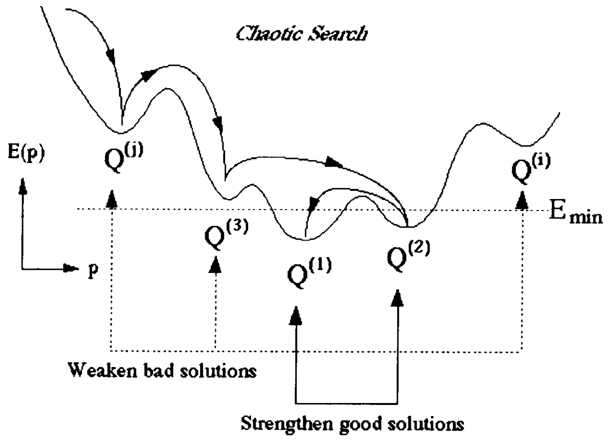


Figure 6.10: Schematic illustration of the essence of the learning algorithm.

Fig. 6.11 shows an example of the learning algorithm applied to random 5-city TSP of fig. 6.1, where the learning parameters are set to $(\mu, \gamma, \theta) = (0, 0.001, 0.00001)$. We see that the chaotic dynamics which makes switches among the optimum solution $Q^{(1)}$ and the second-optimum solution $Q^{(2)}$ gradually converges to the optimum solution. As is also confirmed in fig. 6.12, residence rate of the chaotic dynamics in the optimum state $Q^{(1)}$ eventually increases to 1.0 as the learning proceeds. The experiment therefore shows efficiency of the learning algorithm which modifies asymptotic measure of the chaotic search dynamics to eventually converge to the optimum solution.

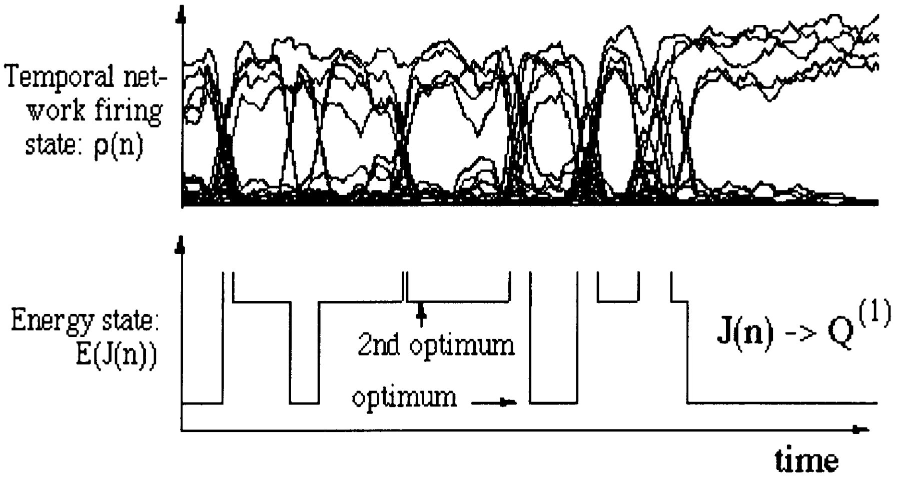


Figure 6.11: Example of the learning algorithm applied to the 5-city TSP. The learning parameters are set to (μ, γ, θ) = (0, 0.001, 0.00001) and other parameters are set same as fig. 6.2.

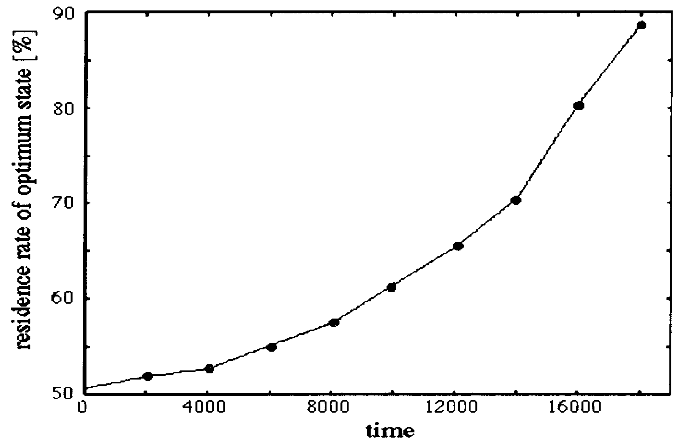


Figure 6.12: Residence rate in the optimum solution with an increasing learning step.

Fig. 6.13 shows result of the learning algorithm applied to a 20-city TSP, where the TSP is selected from the 20 instances of random 20-city TSP of Table 6.2 and the parameters are set to $(r,\ A,\ B,\ \omega,\ \alpha,\ \beta,\ \mu,\ \theta) = (0.7,\ 1.0,\ 1.0,\ 0.75,\ 0.1,\ 0.018,\ 0.01,\ 0)$. By decreasing the learning parameter γ from $\gamma = 2.0 * 10^{-4}$ to $\gamma = 3.0 * 10^{-6}$, averaged annealing solution over 100 sets of random initial conditions is calculated. We see that the residual energy $\delta E = (E_{average} - C_{HK})/C_{HK}$ ($E_{average}$: averaged solution, C_{HK}: Held-Karp lower bound [87]) obtained by the learning algorithm is significantly improved as the convergence time S is increased. This implies that, in the present algorithm, an "infinitely slow" learning provides an optimum result as in the case of the stochastic simulated annealing [71].

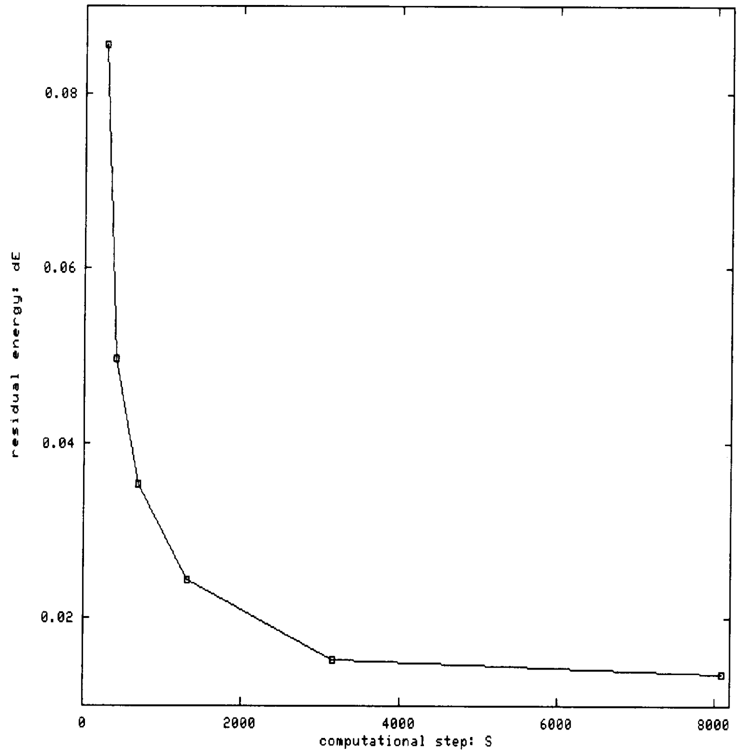


Figure 6.13: Dependence of the residual energy δE against the convergence time S. The residual energy δE is computed from the averaged annealing solution as $\delta E = (E_{average} - C_{HK})/C_{HK}$, where C_{HK} is the Held-Karp lower bound [87] and $E_{average}$ is the averaged solution over 100 sets of random initial conditions.

6.6 Conclusions and discussions

We have analyzed the CSA algorithm in the light of the global bifurcation structure of the chaotic neural networks and reported the limitation of the conventional *slow* CSA algorithm. As an improved algorithm, adaptive CSA algorithm which finds much improved solutions by fast annealing is introduced. Termination condition for the chaotic search is also provided using the Held-Karp lower bound. Since the adaptive CSA algorithm is still an approximate algorithm for finding near-optimum solution, learning algorithm is further introduced as a potential algorithm for realizing simulated annealing in chaotic neural network. The learning algorithm controls the asymptotic measure of the chaotic search dynamics and improves the efficiency of the chaotic search for optimum solution. The experimental study has shown the possibility of realizing convergence of chaotic dynamics to possible global minimum in the limit of *infinitely slow* learning process.

Our present discussions have been based on the application results to relatively small-scale TSPs. Further intensive studies are indispensable to confirm validity of our discussions. Size dependence of the computational step required for the adaptive CSA algorithm will be studied for large-scale problems. Residual energy decreased by slow learning algorithm will be investigated for large-scale problems in our future work. Efficiency of the present algorithms to other combinatorial optimization problems such as the knapsack problem, the quadratic assignment problem would be also examined. By the comparative studies with various other approximate algorithms such as the 2-opt algorithm, the Tabu search algorithm, the genetic algorithm, and many others [124], disadvantages as well as advantages of the CSA algorithms would be clarified.

Chapter 7

Application of chaotic dynamics to nonlinear optimization problems

abstract: Global bifurcation scenario is presented for chaotic dynamical systems that solve nonlinear optimization problems. The bifurcation scenario elucidates the mechanism of chaotic dynamics that searches for a global minimum of the optimization problem. On the basis of the bifurcation scenario, a learning algorithm is introduced for the chaotic optimization system. The algorithm controls the asymptotic measure of the chaotic dynamics and improves efficiency of the "chaotic search." Our numerical experiments also show that the learning algorithm works as "chaotic simulated annealing," which realizes gradual convergence of the chaotic search dynamics to possible optimum solution.

7.1 Introduction

The minimization problem of a nonlinear function that has many local minima appears in many engineering problems. Towards the global minimization of general nonlinear functions, various algorithms have been developed. The famous algorithms are the hill climbing algorithm [27, 72], the function modification algorithm [125, 131], and the simulated annealing algorithm [71]. These algorithms have not yet achieved an applicability to a wide class of practical engineering problems because of the huge computational cost, no guarantee for the global optimality, and the impractically long computational time.

As an alternative approach to the problem, a novel optimization technique of nonlinear functions based on chaotic dynamical systems has been recently developed [65, 66, 181, 187]. In the chaotic optimization technique, the global minimum is searched by chaotic dynamics which visits a variety of local minima of the objective nonlinear function. Although many experimental studies have reported the efficiency of chaos for nonlinear optimization problems, they are mainly based on the simulation studies and therefore they are not sufficient for proving the efficiency of chaos to practical engineering problems. The previous studies also do not clarify the dynamical mechanism of the chaotic optimization. Hence, the present Chapter aims to investigate the chaotic optimization mechanism from the view point of bifurcations in the chaotic optimization system.

First, a global bifurcation scenario is presented for chaotic dynamical systems that solve nonlinear optimization problems. The bifurcation scenario clarifies the generation mechanism of a strange attractor that includes many local minima of the objective nonlinear function as the unstable fixed points. By wandering around a variety of such unstable local minima, the chaotic dynamics searches for the global minimum.

Second, in order to improve the conventional chaotic optimization technique, learning algorithm is introduced to chaotic dynamical system. Our focus is on the asymptotic measure of chaos that determines the efficiency of chaotic optimization. By the learning algorithm, asymptotic measure of the chaotic search is adaptively modified by the history of the chaotic optimization process. By the numerical experiments, efficiency of the learning algorithm for 1- and 2-dimensional nonlinear optimization problems is demonstrated. We also show that the learning algorithm works as a potential algorithm for chaotic simulated annealing (CSA), since it realizes a gradual convergence of the chaotic search dynamics to possible global minimum state in the limit of *infinitely slow* learning.

7.2 Global bifurcation scenario for chaotic optimization system

7.2.1 Chaotic dynamics for function minimization

Consider the following minimization problem of a constrained nonlinear function:

Minimize $e(y)$

Subject to $y \in A$,

where $y = \{y_1, y_2, .., y_d\}$ $(\in R^d)$, $A = \{y : |y_i| < a_i, i = 1, 2, .., d\}$, and $e : A \to R^1$.

Let us construct a chaotic dynamical system that solves the optimization problem. First, diffeomorphism $h : R^d \to A$ is introduced to transform the variables y into new variables $x = \{x_1, x_2, .., x_n\}$ by $y = h(x)$. The objective function $e(y)$ is then redefined as $E(x) = e(h(x))$ for $x \in R^n$. For the diffeomorphism $h = \{h_1, h_2, .., h_n\}$, a set of smooth monotone-increasing functions $h_i : R^d \to R^1$ defined by

$$y_i = h_i(x) = a_i \, \frac{1 - e^{-x_i}}{1 + e^{-x_i}} \quad (i = 1, 2, .., d) \tag{7.1}$$

is exploited. Second, consider the gradient dynamical system

$$\frac{d}{dt} x = - \nabla E(x) \tag{7.2}$$

which exhibits convergence dynamics to local minima of the objective function $E(x)$. Chaotic dynamical system that searches for the global minimum of $E(x)$ is constructed by the Euler's discretization of the gradient system (7.2) as

$$\begin{aligned} x(t+1) &= x(t) - \alpha\, \nabla E(x(t)) \\ &:= f(x(t)), \end{aligned} \tag{7.3}$$

where α stands for the Euler's discretization constant. It is proven by Yamaguti-Matano [215] and Hata [84] that, in certain mathematical condition, the difference equation (7.3) exhibits Li-York chaos [126] for a large enough α. By the global bifurcation analyses of eq. (7.3), the next Section shows the generation mechanism of chaotic dynamics that searches for the global minimum of $E(x)$.

7.2.2 Global bifurcation structure of chaotic search

Using the following 1-dimensional example:

Example 1 (1-dimensional Levy-Montalvo function [125]):

$$\begin{aligned} e(y) &= \frac{\pi}{d}\{B\sin^2(\pi z_1)\sum_{i=1}^{d-1}(z_i - C)^2(1 + B\sin^2(\pi z_{i+1})) + (z_d - C)^2\} + D, \\ & z_i = y_i + 3.5, \quad |y_i| < 5.0, \quad i = 1, 2, .., d, \end{aligned} \tag{7.4}$$

where $d = 1$, $B = 5.0$, $C = 1.0$, and $D = 100$, let us see the generation of chaotic dynamics that searches for the global minimum of $e(y)$.

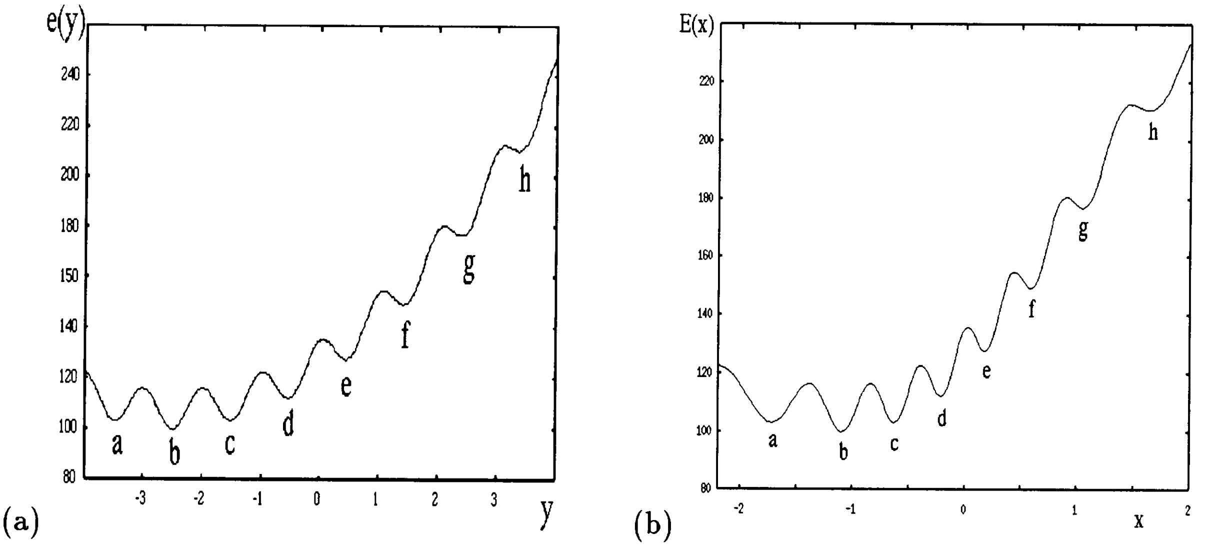


Figure 7.1: (a) Energy curve of the cost function (7.4). **a**, **b**, **c**, **d**, **e**, **f**, **g**, **h** indicate the location of the 8 local minimum solutions. (b) Energy curve of the cost function (7.4) transformed by h.

As is shown by the function curve of fig. 7.1 (a), the function $e(y)$ has eight local minima indicated as **a**, **b**, **c**, **d**, **e**, **f**, **g**, **h**, where **b** ($y = -2.5$) corresponds to the global minimum. Fig. 7.1 (b) shows the objective function curve of $E(x)$ transformed by h with $a = 5$. We see that the original topology of the eight local minima of $e(y)$ is preserved in $E(x)$.

Taking α as the bifurcation parameter, global bifurcation diagram of the difference equation (7.3) constructed for the example 1 is drawn in fig. 7.2.

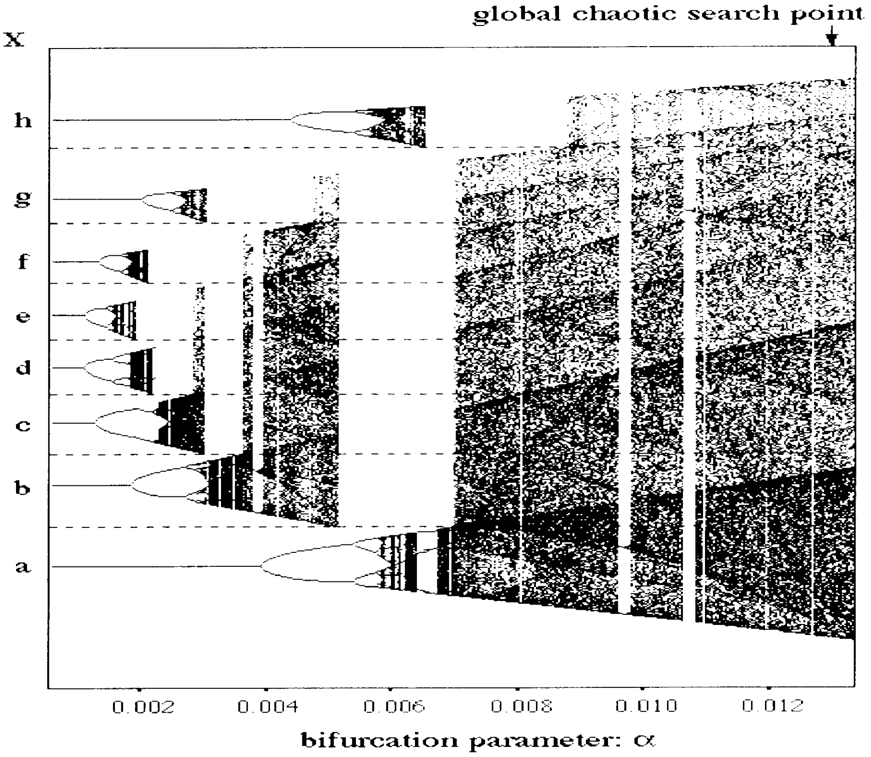


Figure 7.2: Bifurcation diagram of the chaotic dynamical system (7.3) applied to Example 1. With an increase in the bifurcation parameter α, eight bifurcation diagrams are drawn respectively from the local minimum solutions **a**, **b**, **c**, **d**, **e**, **f**, **g**, **h**.

First, in a bifurcation parameter region of small α, there exist eight stable fixed points which correspond to the local minima of the objective function $e(y)$. With an increase in the bifurcation parameter α, each local minimum bifurcates into chaotic attractor through period-doubling bifurcation route to chaos [55]. The chaotic attractors are initially localized in the state space and eventually merge with each other via a series of crises [79, 80]. For instance, the chaotic attractors generated from the local minima **d**, **e**, **f**, **g**, **h** touch the separatrices from the chaotic attractors generated from **c**, **d**, **e**, **f**, **g**, respectively, and loose their stabilities via boundary crises. The chaotic attractors from **a**, **b**, **c** also touch the separatrices from the previous chaotic attractors from **b**, **c**, **d**, respectively, and loose their stabilities via interior crises, where the interior crises give rise to the merger of the localized chaotic attractors. Notice that, via the interior crisis at $\alpha \cong 0.0074$, all of the eight localized chaotic attractors from **a**, **b**, **c**, **d**, **e**, **f**, **g**, **h** merge into a single global chaotic attractor. The single attractor achieves a global dynamical structure which include all the eight local minima as the unstable fixed points. In the sense that the chaotic dynamics on such a global attractor visits the eight unstable local minima successively, the system achieves a search capability of the global minimum among the eight local minima.

7.2.3 Global bifurcation scenario

On the basis of the bifurcation phenomena observed in example 1, we present the following "global bifurcation scenario" for chaotic dynamical systems that solve the optimization problem: First, local minima of $E(x)$ bifurcate into chaotic attractors through period-doubling bifurcation route to chaos. The chaotic attractors are initially localized in the state space and eventually merge into a single global attractor via a series of crises. The final merger gives rise to a single global chaotic attractor which covers all the local minima of $E(x)$. The "chaotic search" for global minimum is realized by chaotic dynamics on such global attractor. The bifurcation scenario is essentially the same with the one presented in Chapter 5. This implies that the present bifurcation scenario elucidates universal bifurcation phenomena that give rise to chaotic optimization dynamics.

One of the major problems for practical application of chaotic optimization technique to various engineering problems is the difficulty of choosing good parameter values of the chaotic search system. In fact, parameter values for the chaotic optimizers have been empirically determined in the conventional studies [65, 154, 181, 187]. For this problem, the present bifurcation scenario provides a clear guideline for tuning the bifurcation parameter value. Namely, by following the bifurcation procedure of local minima of $E(x)$, one may find the bifurcation parameter region where the chaotic attractors from local minima merge into a "global search" attractor.

7.2.4 Chaotic search algorithm

Based on the global chaotic search dynamics at $\alpha = 0.013$, the global minimum of $E(x)$ can be searched by the following algorithm 1.

[Algorithm 1 (Chaotic Search)]

STEP 1 (*Initial condition*):
Choose random initial condition $x(0) \in [-1,1]^d$ and set $t = 0$, $z = E(x(0))$.

STEP 2 (*Update of the dynamics of eq. (7.3)*):

$$x(t+1) = f(x(t))$$

STEP 3 (*Update of the minimum value z*):
Set $z = \min(z, E(x(t+1)))$, $t = t+1$, and go to STEP 2.

By the recursive execution of the STEP 2, chaotic dynamics of eq. (7.3) searches for the global minimum of $E(x)$. Fig. 7.3 shows the result of the algorithm 1 applied to the example 1. As the iteration number of the chaotic search increases, we see that the minimum value z becomes asymptotically close to the global minimum of $E(x)$.

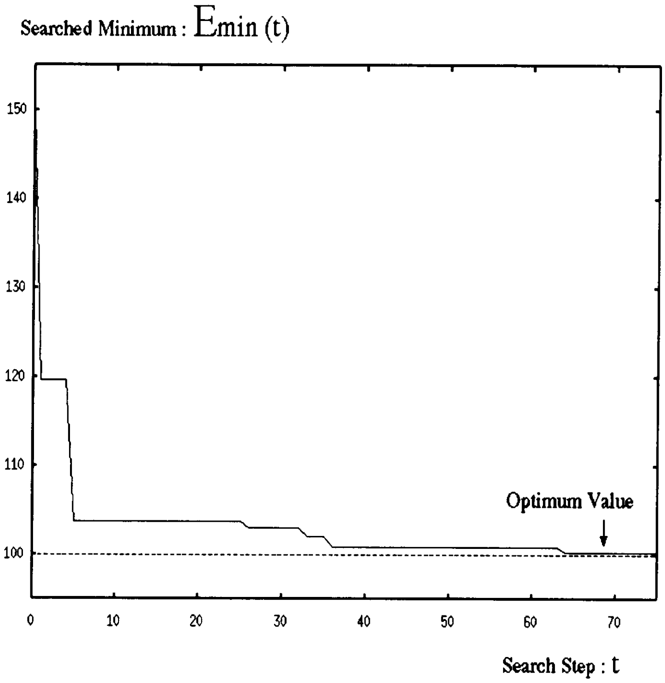


Figure 7.3: Result of the chaotic search algorithm 1 applied to Example 1. The minimum value E_{min} searched by chaotic dynamics is drawn with an iteration step of the algorithm.

In general, chaotic dynamical system has an invariant measure [118] which describes the non-uniform trajectory distribution of the chaotic dynamics. Fig. 7.4 shows an asymptotic measure of the chaotic search dynamics of eq. (7.3) applied to example 1. The measure is computed for a single chaotic trajectory from a random initial condition, where the computational result seems to be independent of the choice of the initial condition $x(0)$. We note that the efficiency of the present chaotic search algorithm is strongly dependent upon this asymptotic measure. The structure of the asymptotic measure is determined by the functional form of the chaotic dynamics and the configuration of the objective function. In the conventional chaotic search techniques [65, 154, 181, 187], methods for constructing an asymptotic measure which gives rise to an efficient chaotic search are not explicitly provided. We remark that this is the limitation of the conventional chaotic search techniques. Systematic methods should be developed for designing the asymptotic measure which has better search capability for the global minimum.

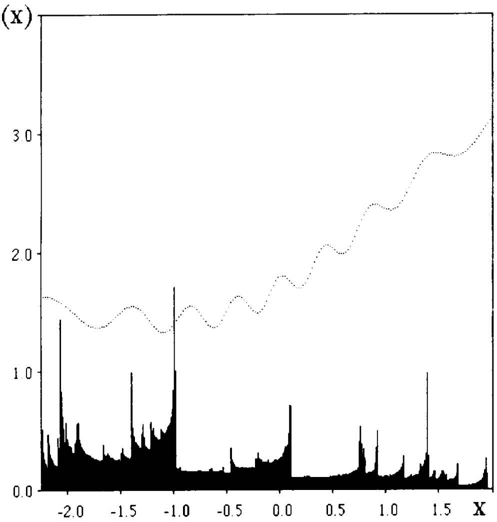


Figure 7.4: Asymptotic measure $\rho(x)$ of the "chaotic search" dynamics (7.3) applied to Example 1.

7.3 Learning algorithm for chaotic search dynamics

In this Section, we introduce a learning algorithm for chaotic search dynamics. Based on the history of the previous chaotic search, the learning algorithm adaptively controls the asymptotic measure to realize better chaotic search for optimum solution [194].

7.3.1 Continuous piecewise-linear filter

Let us introduce a continuous piecewise-linear (CPL) filter $g = \{g_1, g_2, .., g_d\}$, whose i-th component is described by a CPL map g_i of fig. 7.5. The i-th CPL map g_i has a range $J^i = [b^i, c^i]$, which is uniformly divided into N unit ranges $J^i_j = [b^i + (j-1)D^i,\ b^i + jD^i]$ ($D^i = (c^i - b^i)/N$, $j = 1, 2, .., N$), and a domain $I^i = [b^i, c^i]$, which is non-uniformly divided into N unit domains $I^i_j = [b^i_j, c^i_j]$ ($j = 1, 2, .., N$). Here, the length of each unit domain $d^i_j = |c^i_j - b^i_j|$ is set to be varied from others, where each length is determined as $d^i_j = \omega^i_j(c^i - b^i)$ using a set of weight parameters $\{\omega^i_j : j = 1, 2, .., N\}$ that satisfies the normalization condition $\sum_{j=1}^{N} \omega^i_j = 1$ ($\omega^i_j > 0$).

Suppose there is a set of input trajectories to g_i with uniform distribution. By the non-uniformity of g_i, the output trajectories are non-uniformly distributed according to the weight parameters $\{\omega^i_j\}$. For instance, if the k-th parameter ω^i_k has a large weight, the length d^i_k of the k-th input domain I^i_k is set to be large and the trajectory distribution on the k-th output range J_k^i increases. By using the CPL filter g which creates a non-uniformity in the trajectory distribution, asymptotic measure of the chaotic dynamics can be controlled.

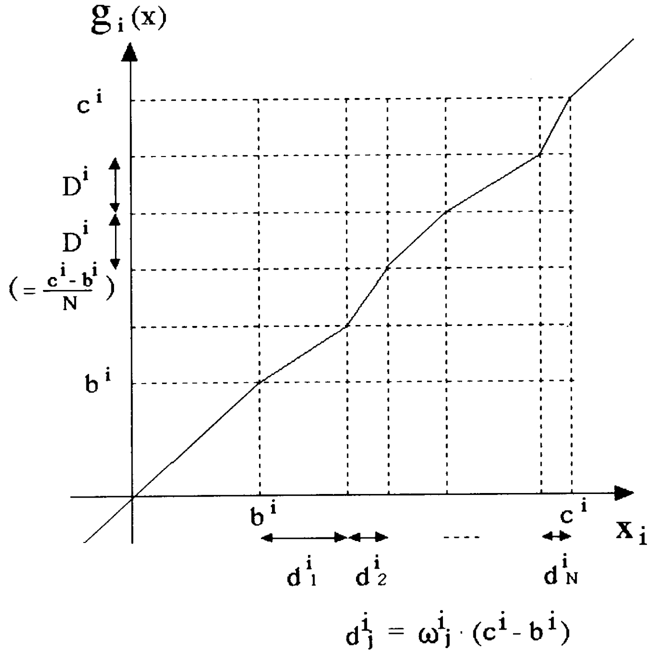


Figure 7.5: Basic structure of the continuous piecewise linear map g_i. g_i has a range $J^i = [b^i, c^i]$ which is uniformly divided into N range blocks and a domain $I^i = [b^i, c^i]$ which is non-uniformly divided into N domain blocks.

7.3.2 Learning algorithm

By introducing the CPL-filter g as the feedback filter of the chaotic search dynamics, let us consider the following learning algorithm.

[Algorithm 2 (Learning Chaotic Dynamics)]

STEP 1 (*Initial condition*):
Choose random initial condition $x(0) \in [-1,1]^d$ and set $t = 0$ and $z = E(x(0))$. Set the weight parameter values as $\omega_1^1 = \omega_2^1 = \cdots = \omega_d^N = 1/N$, that make the CPL-filter an identity map $g = i.d.$

STEP 2 (*Update of the feedback filtered dynamics*):

$$x(t+1) = f(g(x(t))). \tag{7.5}$$

STEP 3 (*Learning in chaotic dynamics*):
If $E(x(t+1)) \leq (1+\epsilon)z$, then, for each CPL-map g_i, increase the j-th weight value that satisfies $x_i(t) \in I_j^i$ by $\omega_j^i = (1+\beta)\omega_j^i$.
If $E(x(t+1)) > (1+\epsilon)z$, then, for each CPL-map g_i, decrease the j-th weight value that satisfies $x_i(t) \in I_j^i$ by $\omega'^i_j = (1-\gamma)\omega_j^i$.
(ϵ ($\epsilon > 0$) stands for a small constant parameter and (β, γ) ($\beta, \gamma > 0$) stands for a set of learning parameters)

STEP 4 (*Modification of CPL-filter*):
Modify the structure of each CPL-map g_i according to i-th set of non-uniform weight parameters that satisfy $\sum_{j=1}^{N} \omega_j^i = 1$. Then, set $n = n+1$, $z = \min(z, E(x(t+1)))$, and $t = t+1$, and go to STEP 2.

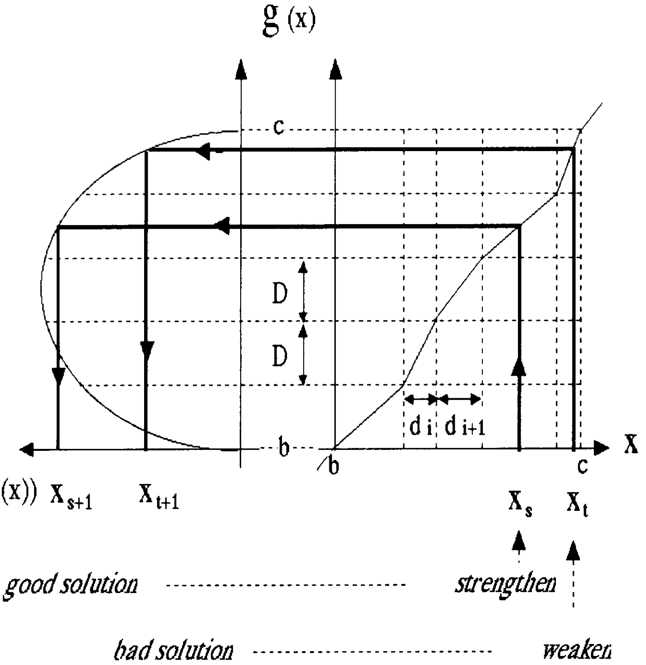


Figure 7.6: Schematic illustration of the learning algorithm 2. Feedback filter g is introduced for chaotic dynamical system (7.3). By adjusting the feedback filter, asymptotic measure of the chaotic search dynamics around the bad solution is decreased, while the asymptotic measure around the good solution is increased.

In the STEP 3 of the algorithm 2, learning process is introduced. As is schematically shown in fig. 7.6, the learning algorithm has the following functions.

1. Decrease the distribution of chaotic search trajectories near the bad solutions. This creates the "Tabu" search area in the asymptotic measure of the chaotic dynamics.

2. Increase the distribution of chaotic search trajectories near good solutions. This gives rise to the local intensive search around good solutions.

7.3.3 Experiment

In order to test the performance of the learning algorithm, let us apply the algorithm 2 to the example 1. The results are shown in figs. 7.7 (a)-(d), which display successive changes in the asymptotic measure of the chaotic search of eq. (7.3) by the learning algorithm. The parameters of the algorithm are set as $(b,\ c,\ N,\ \epsilon,\ \beta,\ \gamma) = (-2.25,\ 2.0,\ 20,\ 0.025,\ 0.025,\ 0.0005)$. As the learning proceeds, we see that the asymptotic measure around the optimum solution grows high and after the 1000-iterative learning the asymptotic measure eventually converges to the optimum solution. This demonstrates the function of the learning algorithm, which eventually narrows the chaotic search region and realizes a final convergence to the optimum solution. The result of the learning algorithm seem to be independent of the choice of the random initial condition.

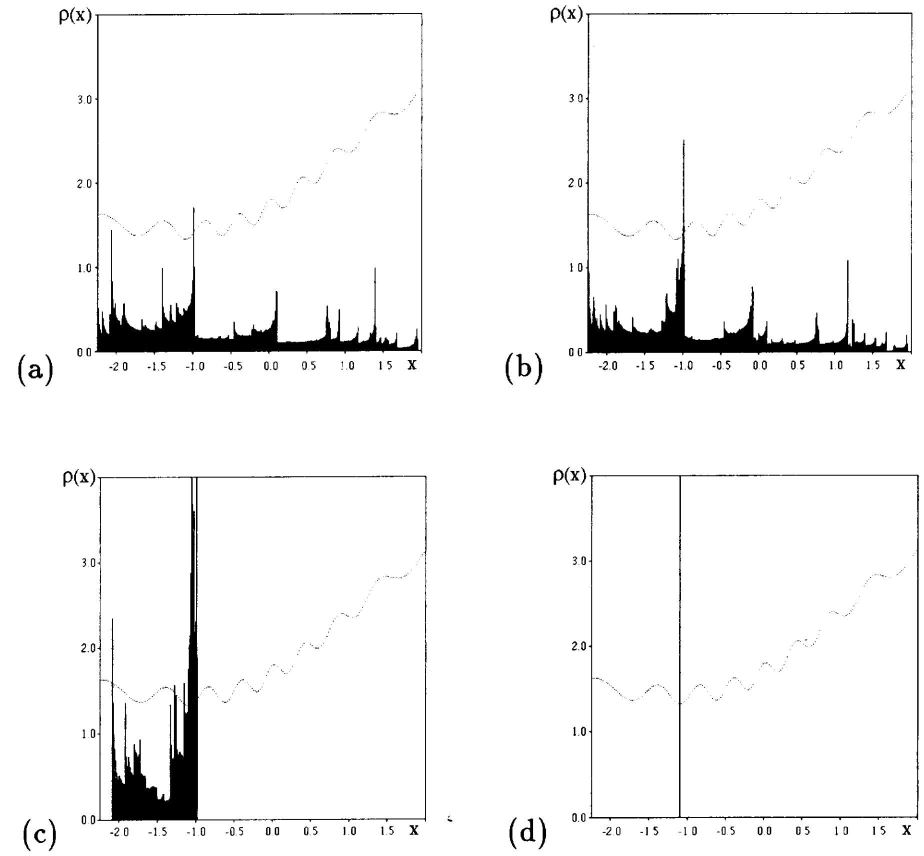


Figure 7.7: Asymptotic measure $\rho(x)$ of the "chaotic search" dynamics applied to Example 1. (a) Before learning. (b) After 1000-iterative learning. (c) After 1200-iterative learning. (d) After 1800-iterative learning.

In Table 7.1, performance of the algorithm 1 (without learning) and the algorithm 2 (with learning) is compared by the averaged iteration steps to reach to the global minimum. By the learning algorithm, which effectively forms a Tabu search area, average time to reach to the global minimum is improved.

7.3.4 Another 1-dimensional example

Let us consider another 1-dimensional example.

Example 2:

$$\begin{aligned} e(y) &= 600 + (y+4.2)\,(y+2.2)\,(y+1.2)\,(y-1.7) \\ &\quad (y-2.8)\,(y-3.865) \quad (|y| < 4.3). \end{aligned} \tag{7.6}$$

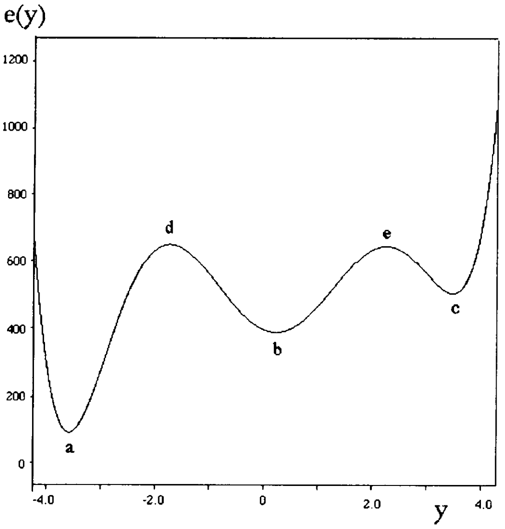


Figure 7.8: Energy curve of the cost function (7.6). **a**, **b**, **c** correspond to the local minimum solutions, whereas **d**, **e** correspond to the local maximum solutions.

As is shown in fig. 7.8, the energy curve of eq. (7.6) has three local minima **a**, **b**, **c**, where **a** corresponds to the global minimum. Fig. 7.9 shows the bifurcation diagram of the dynamical system of eq. (7.3) ($a = 4.3$) for the example 2. With an increase in the bifurcation parameter α, bifurcations of the local minima **a**, **b**, **c** similar to the example 1 are discernible. First, the local minima **a**, **b**, **c** bifurcate into chaotic attractors through period-doubling bifurcation route to chaos. The chaotic attractors are initially localized around the corresponding local minima and eventually merge with each other via crises. The interior crisis near $\alpha = 0.008$ gives rise to the final merger of all the localized chaotic attractors.

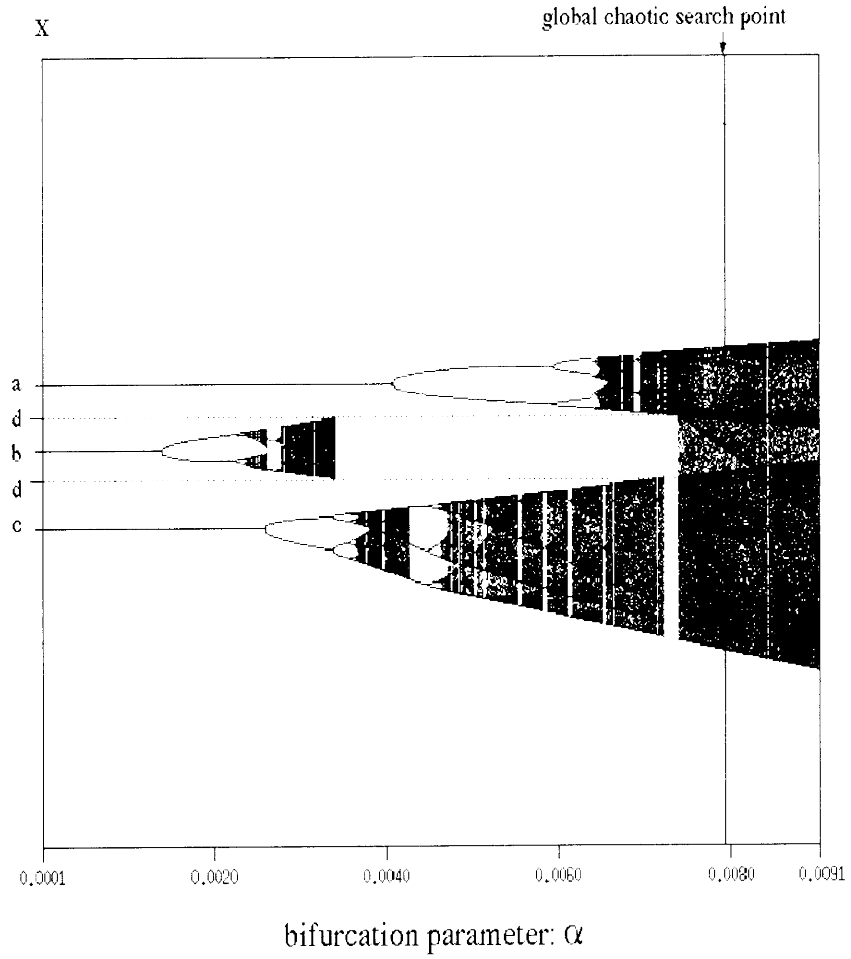


Figure 7.9: Bifurcation diagram of the chaotic dynamical system (7.3) applied to Example 2. With an increase in the bifurcation parameter α, three bifurcation diagrams are drawn respectively from the local minimum solutions **a**, **b**, **c**.

For the global chaotic search dynamics with $\alpha = 0.008$, the learning algorithm is applied. Figs. 7.10 show the application results, where the parameters of the learning algorithm are set as $(b, c, N, \epsilon, \beta, \gamma) = (-6.5, 3.5, 50, 0.1, 0.1, 0.01)$. Similar results to the example 1 can be observed in figs. 7.10. Namely, as the learning proceeds, the asymptotic measure around the optimum solution grows high and after the 1000-iterative steps of learning the asymptotic measure eventually converges to the optimum solution. Comparative study of the algorithms 1 and 2 also shows that the average time to search for the global minimum is shortened by the learning algorithm (see Table 7.1).

For 1-dimensional instances other than the examples 1 and 2, we can confirm efficiency of the learning algorithm in a similar manner.

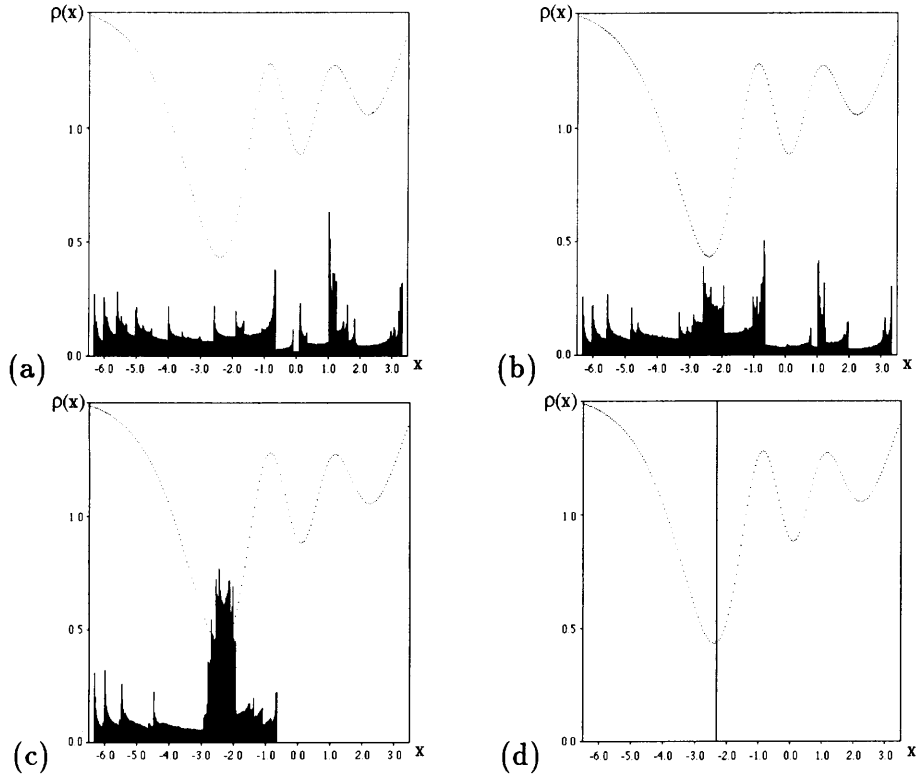


Figure 7.10: Asymptotic measure $\rho(x)$ of the "chaotic search" dynamics applied to Example 2. (a) Before learning. (b) After 500-iterative learning. (c) After 600-iterative learning. (d) After 700-iterative learning.

7.3.5 2-dimensional problems

Example 3 (2-Dimensional Levy-Montalvo Function [125]):
Here we consider the 2-dimensional case of the Levy-Montalvo function of eq. (7.4), where the parameters are set as $d = 2$, $B = 10$, $C = 1.0$, $D = 20.0$.

As is shown in fig. 7.11, the 2-dimensional Levy-Montalvo function has many local minima, where the unique global minimum is located at $(y_1, y_2) = (-2.5, -2.5)$.

For the 2-dimensional example, the dynamical system of eq. (7.3) is slightly modified as $x_i(t+1) = x_i(t) - \alpha \partial E / \partial x_i(x(t)) + \sigma(x_i(t))$ $(i = 1, 2)$, where the additional term is given by $\sigma(x) = q_1 + q_2 \exp(-r_2(x - s_2)^2)$ (if $x \leq s_1$), $q_1 \exp(-r_1(x - s_1)^2) + q_2 \exp(-r_2(x - s_2)^2)$ (if $s_1 < x < s_2$), $q_1 \exp(-r_1(x - s_1)^2) + q_2$ (if $s_2 \leq x$). For the parameters set to $(a, \alpha, q_1, q_2, r_1, r_2, s_1, s_2) = (5, 0.00075, 1, -1, 10, 10, -3, 3)$, the modified system exhibits chaotic search dynamics which includes many local minima in the global dynamical structure (see fig. 7.12 (a)).

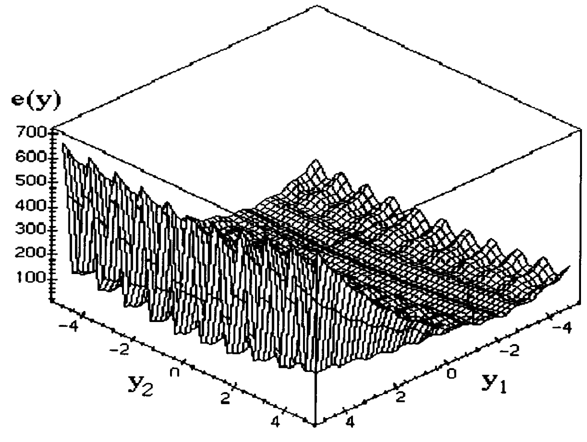


Figure 7.11: Energy curve of the 2-dimensional Levy-Montalvo function (7.4). In the domain $A = \{y : |y_1| < 5.0, |y_2| < 5.0\}$, there exist many local minimum solutions and a single global minimum at $(y_1, y_2) = (-2.5, -2.5)$.

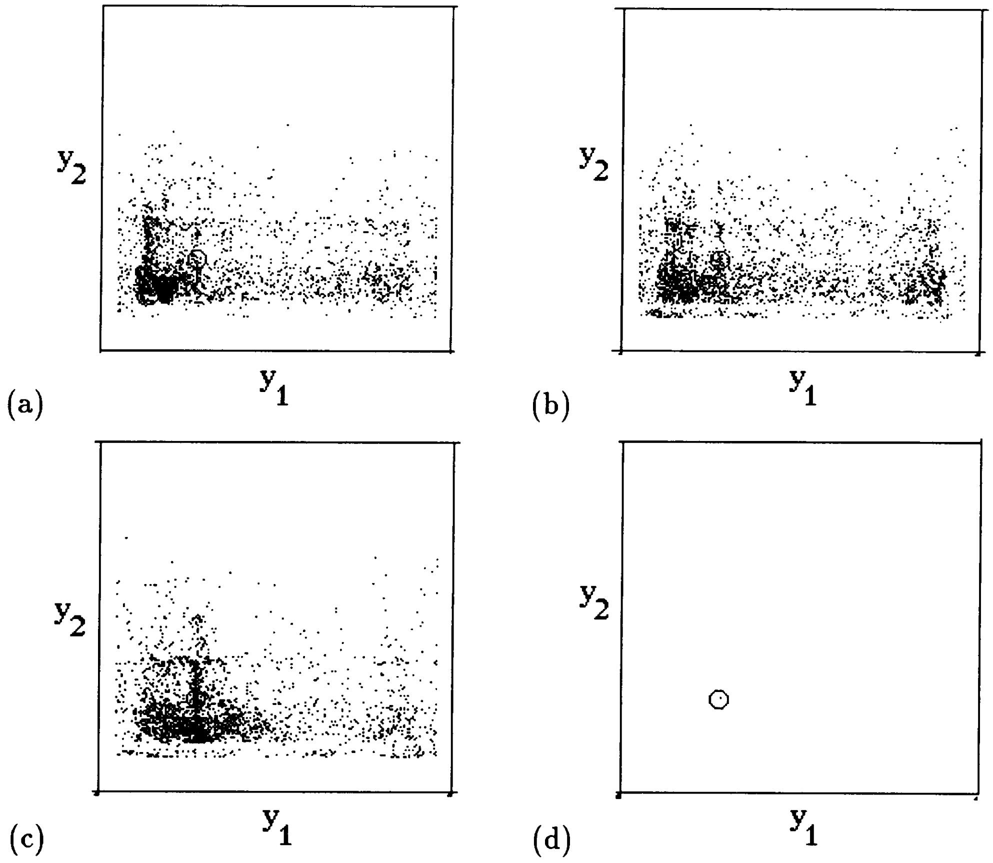


Figure 7.12: Trajectory distribution of the chaotic search dynamics applied to Example 3. The circle indicates the existence region of the optimum solution. (a) Before learning. (b) After 200000-iterative learning. (c) After 400000-iterative learning. (d) After 600000-iterative learning.

For this global chaotic search dynamics, learning algorithm is applied. The learning parameters are set as $(b_1, b_2, c_1, c_2, N, \epsilon, \beta, \gamma) = (-2.67, -19.5, 1.42, 23.5, 200, 0.005, 0.005, 0.00001)$ and the results are shown in figs. 7.12 (a)-(d).

Initially, the chaotic dynamics gives rise to global search, whose trajectories are distributed globally in the 2-dimensional state space (see fig. 7.12 (a)). As the learning proceeds, asymptotic measure of the chaotic dynamics increases around the global minimum (see figs. 7.12 (b),(c)) and eventually converges to the global minimum (see fig. 7.12 (d)). This implies that the learning algorithm works efficiently for the 2-dimensional example. Comparative study of the algorithms 1 and 2 also shows the average search time for the global minimum shortened by the learning algorithm (see Table 7.1).

Example 4 (Girewank Function [72]):

$$\begin{aligned} e(y) &= -\cos(y_1)\cos(\frac{y_2}{\sqrt{2}}) + \frac{y_1^2 + y_2^2}{200} + 21, \\ & |y_i| < 25.0, \quad i = 1, 2. \end{aligned} \tag{7.7}$$

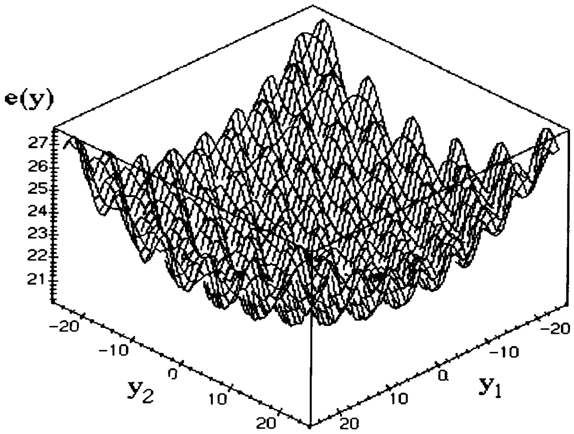


Figure 7.13: Energy curve of the Girewank function (7.7). In the domain $A = \{y : |y_1| < 25.0, |y_2| < 25.0\}$, there exist many local minimum solutions and a single global minimum at the origin $(y_1, y_2) = (0.0, 0.0)$.

As is shown in fig. 7.13, the Girewank function of eq. (7.7) has many local minima, where the unique global minimum is located at the origin $(y_1,y_2)=(0,0)$.

For the parameters set to $(a, \alpha, q_1, q_2, r_1, r_2, s_1, s_2) = (25, 0.3, 3, -3, 10, 10, -3, 3)$, the dynamical system of modified eq. (7.3) exhibits global chaotic search dynamics which includes many local minima in the global dynamical structure (see fig. 7.14 (a)).

Figs. 7.14 (a)-(d) show the results of the learning algorithm applied to the example 4, where the learning parameters are set to $(b_1, b_2, c_1, c_2, N, \epsilon, \beta, \gamma) = (-4.2, -3.2, 4.2, 3.2, 200, 0.02, 0.02, 0.0005)$. Initially, the chaotic dynamics exhibits global search dynamics, which is uniformly distributed in the 2-dimensional space (see fig. 7.14 (a)). As the learning proceeds, the chaotic dynamics achieves a non-uniform (torus-breakdown type) dynamical structure and its asymptotic measure increases around the global minimum (see figs. 7.14 (b),(c)). Finally, the system converges to the global minimum (see fig. 7.14 (d)). Efficiency of the learning algorithm for the example 4 can also be confirmed by the average search time of the global minimum in Table 7.1.

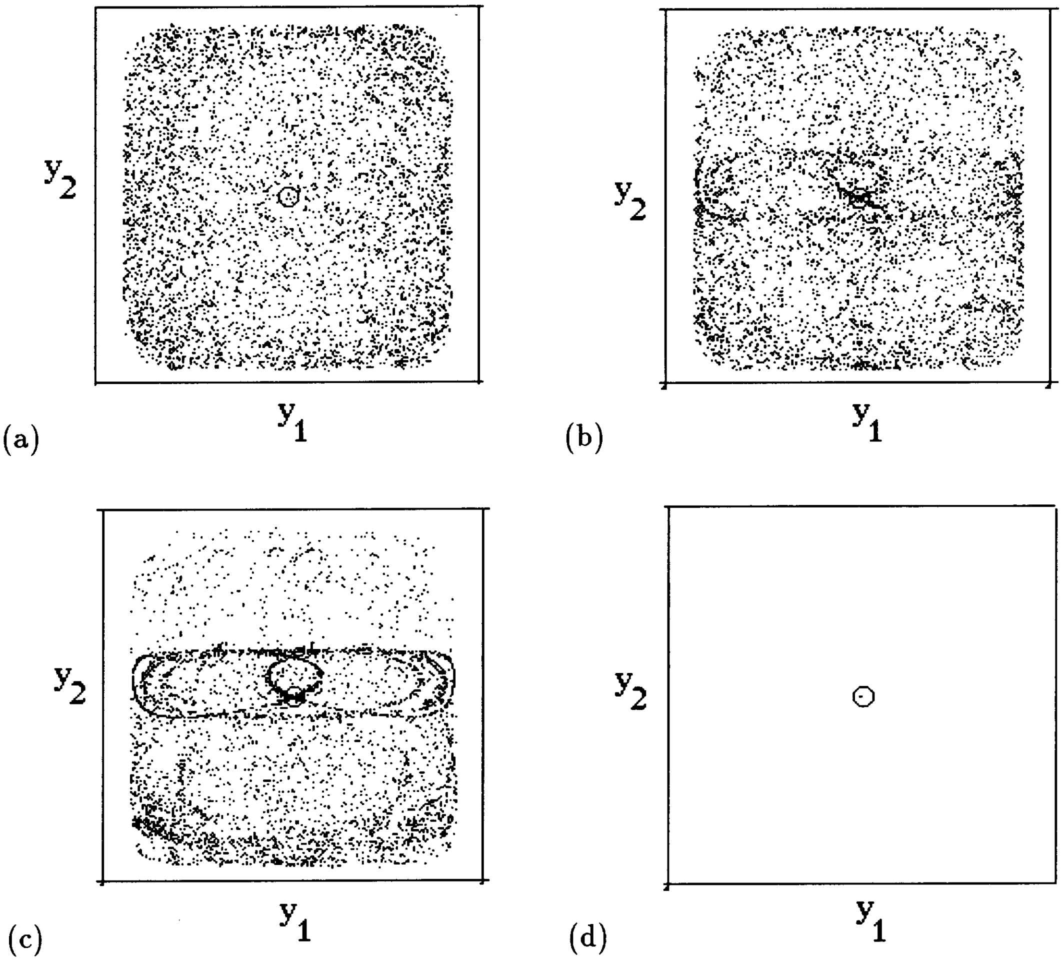


Figure 7.14: Trajectory distribution of the chaotic search dynamics applied to Example 4. The circle indicates the existence region of the optimum solution. (a) Before learning. (b) After 200000-iterative learning. (c) After 400000-iterative learning. (d) After 600000-iterative learning.

Table 7.1: Results of the chaotic search algorithms 1 and 2 applied to Examples 1-4. The average iteration steps to reach to global minimum solution is calculated for 10000 sets of random initial conditions.

Example	*Without Learning*	*With Learning*
Example 1	406.7 ± 0.8	328.1 ± 0.7
Example 2	815.0 ± 1.0	411.0 ± 1.0
Example 3	11816 ± 100	8726 ± 80
Example 4	24614 ± 150	18117 ± 100

7.3.6 Learning as simulated annealing

From the numerical experiments with four instances of nonlinear function minimization problems, we have seen the efficiency of the learning algorithm which gradually narrows the chaotic search space and finally makes the chaotic dynamics to converge to the global minimum. For the stochastic dynamical system, the search space is usually controlled by the *simulated annealing algorithm* [71], which eventually narrows the stochastic search space and realizes a final convergence to the global minimum by the thermodynamical control of the temperature parameter. According to the way of controlling the search space, the present learning algorithm can be also considered as a novel simulated annealing algorithm for chaotic dynamical system.

As a simulated annealing algorithm for chaotic dynamical system, chaotic simulated annealing (CSA) algorithm has been proposed by Chen & Aihara [35]. The conventional CSA algorithm is based on the controlling algorithm of a single bifurcation parameter in chaotic search system. By gradually changing the bifurcation parameter, the chaotic search dynamics is eventually changed into the gradient descent dynamics which makes final convergence to possible global minimum. As is discussed in Chapter 6, the result of the CSA algorithm is primarily dependent upon the global bifurcation structure of the chaotic dynamical system and global optimality of the CSA algorithm is not always guaranteed. In fact, for the four instances of nonlinear optimization problems studied in this Chapter, global minimum solution can not be obtained by an *infinitely slow* CSA algorithm. Hence the experimental results of the present study implies a possibility that the difficulties of the conventional CSA algorithm can be conquered by the learning algorithm of the chaotic dynamical systems.

7.4 Conclusions and discussions

A global bifurcation scenario is presented for chaotic dynamical systems that solve nonlinear optimization problems. The scenario well elucidates the chaotic search mechanism that visits all local minima of the object function as the unstable fixed points.

In order to improve efficiency of the chaotic optimization algorithm, CPL feedback filter and its learning algorithm are then introduced. The learning algorithm adaptively controls the asymptotic measure of the chaotic dynamics and realizes an efficient search for optimum solution. Using four instances of 1- and 2-dimensional nonlinear optimization problems, efficiency of the algorithm has been demonstrated.

We remark that there are only few studies which deal with the learning in chaotic dynamical system. To the best of our knowledge, learning in chaotic dynamics has been discussed by Tsuda [203], Nara *et al.* [150, 151], Watanabe *et al.* [207] for the applications to pattern learning, pattern search, and memory acquisition. We believe that the present study provides a first result for applying the learning algorithm in chaotic dynamics for the optimization problems.

Followings are our future problems.

1. Efficiency of the learning algorithm is tested for higher dimensional problems.

2. By the numerical experiments, we have seen that the learning algorithm works as a simulated annealing algorithm which gradually cools down the chaotic search dynamics to converge to the global minimum. As in the stochastic simulated annealing [71], "gradual" learning might also be necessary for the global optimization of the chaotic dynamics. Fig. 7.15 shows the dependence of the convergence rate to the global minimum on the learning speed β. For a large β which realizes a fast learning, the trapping rate to local minima increases. On the other hand, for a small β which slows down the learning process, global optimization rate of almost 100% is realized. This implies that the learning parameter provides a trade-off parameter between the global optimization rate and the learning speed. Guideline should be further developed for determining a good learning parameter value in order to realize global optimization by fast learning.

3. In the recent studies of chaos engineering, techniques have been sought for controlling chaotic dynamics in the way of harnessing the wild horses. Towards the "harnessing of chaos" [214], many control techniques such as the OGY method [155] have been developed. In the conventional techniques, the local dynamical structure of chaos is mainly controlled to stabilize the periodic dynamics. In the sense that the present learning algorithm controls the global dynamical structure of the asymptotic measure of chaos, the learning algorithm may provide us with more adaptive technique for controlling chaos. Applicability of the learning algorithm for the chaos control technique would be also studied in our future works.

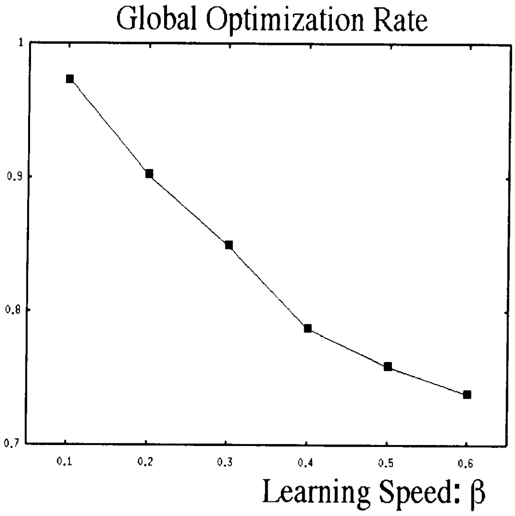


Figure 7.15: Dependence of the global optimization rate of the learning algorithm to the learning speed β. The global optimization rate is calculate for 5000 sets of random initial conditions.

Chapter 8

Conclusion

8.1 Summary

In this thesis, we have studied the functions of chaotic dynamics in neural information processing systems.

In Chapter 2, we have considered the learning capability of neural networks with time delayed feedbacks. The experimental studies have shown that the time delays drastically enhance the learning capability of the neural networks especially in case of learning high-dimensional chaotic dynamics.

In Chapters 3 and 4, we have developed an algorithm for constructing a parametrized family of neural networks that identify the underlying bifurcation parameters of chaotic time series. The family of neural networks can be used for recognition of chaotic time series and for detection of switch dynamics in chaotic time series.

We emphasize that these applications are possible only when the neural networks learn and exhibit chaotic dynamics. Since chaotic dynamics provides us with a rich information on its global dynamical structure, learning capability of the neural networks is enhanced and identification of the underlying bifurcation parameters of chaotic dynamics becomes possible.

In Chapters 5-7, we studied the optimization capability of chaotic neural networks that exhibit chaotic dynamics to search for optimum or near-optimum solution of the optimization problem. A global bifurcation scenario is presented to elucidate the chaotic search mechanism that efficiently escapes from local minima and searches for another better solution. On the basis of the bifurcation scenario, difficulty of realizing simulated annealing in chaotic dynamical system is discussed and several modified algorithms for chaotic simulated annealing are developed.

As is summarized above, we have seen many applications in which chaotic dynamics significantly enhances the information processing capability of the neural networks. Our studies therefore encourage further applications of the idea of chaotic dynamics to neural networks in a variety of engineering prob-

lems. We also believe that our studies would provide us with deeper insights on the function of chaos in the real brain systems.

8.2 Future works and applications

8.2.1 Chaotic memory in delayed neural networks

In Chapter 2, we have studied a learning capability of neural networks with time-delayed feedbacks. Such delayed feedback systems described by retarded functional differential equations can be found in many engineering systems. One example is the optical laser system [99], which is known to give rise to multi-stability of infinitely many periodic attractors. With a change in bifurcation parameter, higher harmonic bifurcations take place and the infinitely many periodic attractors eventually merge into a single global chaotic attractor [99]. This multi-stability and their merging bifurcations have practical applicability to memory device, in which memories are stored in the periodic attractors and the stored memories are searched by global chaotic dynamics [4]. It is an easy application for delayed neural networks to model such laser dynamics that realizes chaotic memory function, since both models are described by retarded functional differential equations. Hence, it is our future work to implement chaotic memory for delayed neural networks and apply the neural memory device for *multiple oscillatory mode generation* and for *temporal data storage.*

8.2.2 Application to real world problems

In Chapters 2-4, we have studied the neural network applications of modeling chaotic dynamics and identifying a parametrized family of chaotic dynamics. Although the neural networks efficiently work for learning equational models of chaotic dynamics, its applicability to chaotic dynamics in real world systems has not yet been tested. One of our important future works is to apply our algorithm to real world dynamical systems such as electronic circuits [135], chaotic chemical sensor [169], blast furnace [144], flooded ship motion [146], and chaotic neural oscillator [9, 134].

A. System control and monitoring

There are many real engineering systems such as chemical sensor [169] and blast furnace [144], whose equation of motion and bifurcation structure are not well understood but whose dynamics can be observed as time series data. Identification of the global bifurcation structure of such unknown dynamical system from time series data by the algorithm of Chapter 3 may provide us with a rich information on the original dynamical structure, which can be utilized for *control* and *monitoring* of chaotic engineering systems.

B. Testing plausibility of qualitative models

There are also many real world systems such as physiological neuron [9, 134] and flooded ship dynamics [146], whose bifurcation structures have been analyzed by using qualitative mathematical models. It is an interesting future work to *test plausibility* of the qualitative mathematical models from the comparative study between the bifurcation structure of the qualitative mathematical models and the bifurcation structure reconstructed from time series data by the algorithm of Chapter 3.

C. Judging low-dimensional chaos in time series

In nonlinear analysis of complex time series observed form real world systems, the question often arises as

Does the irregularity observed in time series data originate from low-dimensional deterministic chaos or infinite-dimensional stochastic dynamics?

In conventional techniques of chaotic time series analyses [1, 34, 101, 109, 189, 208], the question has been tried to be answered by detecting low-dimensional chaotic property in time series data. This type of approach can be sometimes controversial, since accurate characterization of chaotic property becomes very difficult when time series data is too short and when time series data is contaminated with too strong observational noise or dynamical noise.

We emphasize that our approach of identifying a parametrized family of chaotic dynamics from time series data can solve this problem. If low-dimensional bifurcation structure is clearly detected by our algorithm, that is the *strongest evidence* for low-dimensional chaos in time series data, because stochastic dynamics never gives rise to such low-dimensional bifurcations. Hence, our algorithm may be further utilized to *judge low-dimensional chaos* in a variety of real world time series that have been miss-judged as stochastic dynamics.

D. Difficulty of modeling real world dynamics

Finally, we note that application of the algorithms of Chapters 2-4 to real world systems may not be in some cases carried out straightforwardly, since the real world systems often give rise to higher-dimensional dynamics and they also have spatio-temporal structure [40]. In conventional techniques for modeling chaotic dynamics, the delay-coordinate method is usually utilized to reconstruct chaotic dynamics. Although the Takens theorem [186] guarantees that the delay-coordinate method provides an embedding for low-dimensional dynamics, the theorem can not be usually applied to higher dimensional spatio-temporal dynamics. Therefore, the algorithm should be further modified in order to deal with such complex real world dynamics.

8.2.3 Further development and application of chaotic search

A. Design and control of chaotic search

In Chapters 5-7, we have studied the mechanism of chaotic search dynamics in optimization problems. It should be noted that the chaotic optimization capability is primarily dependent upon the asymptotic measure of the chaotic search dynamics. In the conventional chaotic optimization techniques [65, 154, 181, 187], methods for constructing an asymptotic measure which gives rise to an efficient chaotic search are not explicitly provided. We remark that this is the limitation of the conventional chaotic optimization algorithms. In order to realize more efficient chaotic optimization, methods for constructing an asymptotic measure that gives rise to an efficient chaotic search should be further developed. In general, however, construction of chaotic dynamics with desirable asymptotic measure is an extremely difficult engineering problem. At present, there exists a method for constructing a desirable asymptotic measure only for one-dimensional piecewise-linear Markov maps [117]. It is a challenging but worthwhile investigation to develop methods for *designing the asymptotic measure of high-dimensional chaotic neural networks* that significantly improve the chaotic optimization capability.

B. Learning algorithm for simulated annealing

Construction of simulated annealing algorithm in chaotic dynamics is also an important and interesting open problem. Our studies of Chapters 6 and 7 imply possibility of *realizing a simulated annealing in chaotic dynamics by introducing a learning algorithm*, since the learning gradually narrows the chaotic search space and finally makes the chaotic search dynamics to converge to a possible global minimum. Our further study will be devoted to justify this scenario from experimental and theoretical viewpoint.

C. Hardware implementation of learning

We also note that the learning algorithm introduced to chaotic neural network in Chapter 6 is based upon a *global* learning rule in the sense that the synaptic connections are adjusted by computing the energy function defined to *global* dynamical state of the chaotic neural network. Observation of the global network state at every learning step is computationally rather costly. Development of the learning algorithm based *local* rule, which is to adjust the synaptic connections by observing only their neighboring neuron states, may solve this problem. The local learning rule may also be suitable for a hardware implementation of the learning algorithm. Towards possible *hardware implementation* of the learning algorithm in chaotic neural network, *local* learning rule will be developed in our future work.

Bibliography

[1] H. D. I. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, "The analysis of observed chaotic data in physical systems," Review of Modern Physics **65** **(4)** (1993) 1331–1392.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," Cognitive Science **9** (1985) 147–169.

[3] M. Adachi and K. Aihara, "Associative dynamics in a chaotic neural network," *Neural Networks* **10** **(1)** (1997) 83–98.

[4] T. Aida and P. Davis, "Oscillation modes of laser diode pumped hybrid bistable system with large delay and application to dynamical memory," IEEE Journal of Quantum Elecronics **28** (1992) 686–699.

[5] K. Aihara, "Chaotic neural network," In H. Kawakami, Ed., *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems* (World Scientific, Singapore, 1990) pp. 143–161.

[6] K. Aihara, Ed., *Chaos in Neural Systems*, (Tokyo Denki University Press, 1993).

[7] K. Aihara, Ed., *Applied Chaos*, (Science-sha, 1994).

[8] K. Aihara, M. Kotani, and G. Matsumoto, "Chaos and bifurcations in dynamical systems of nerve membranes," In P. L. Christiansen and R. D. Parmentier Eds., *Structure, Coherence and Chaos in Dynamical Systems* (Manchester University Press: Manchester, 1989) pp. 613–617.

[9] K. Aihara and G. Matsumoto, "Chaotic oscillations and bifurcations in squid giant axons," In A. V. Holden Ed., *Chaos* (Manchester University Press: Manchester, Princeton University Press: Princeton, 1986) pp. 257–269.

[10] K. Aihara, G. Matsumoto, and Y. Ikegaya, "Periodic and non-periodic responses of a periodically forced Hodgkin-Huxley oscillator," Journal of Theoretical Biology **109** (1984) 249–269.

[11] K. Aihara, T. Numajiri, G. Matsumoto, and M. Kotani, "Structures of attractors in periodically forced neural oscillators," Physics Letters A **116** **(7)** (1986) 313–317.

[12] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," Physics Letters A **144** (1990) 333-340.

[13] S. V. B. Aiyer, M. Niranjan, and F. Frank, "A theoretical investigation into the performance of the Hopfield model," IEEE Transactions on Neural Networks **1 (2)** (1990) 204-215.

[14] A. M. Albano, J. Muench, C. Schwartz, A. I. Mees, and P. E. Rapp, "Singular-value decomposition and the Grassberger-Procaccia algorithm," Physical Review A **38 (6)** (1988) 3017-3026.

[15] S. Amari, "Characteristics of random nets of analog neuron-like elements," IEEE Transactions on Systems, Man, and Cybernetics **SMC-2** (1972) 643-657.

[16] S. Amari, "Learning patterns and pattern sequences by self-organizing nets of threshold elements," IEEE Transactions on Computers **C-21** (1972) 1197-1206.

[17] S. Amari and K. Maginu, "Statistical neurodynamics of associative memory," Neural Networks **1** (1988) 63-73.

[18] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing infinite numbers of patterns in a spin-glass model of neural networks," Physical Review Letters **55** (1985) 1530-1533.

[19] C. H. Anderson and D. C. van Essen, "Shifter circuits," Proceedings of the National Academy of Sciences, USA **84** (1987) 6297-6301.

[20] J. A. Anderson, "A memory model using spatial correlation functions," Kybernetik **5** (1968) 113-119.

[21] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis* (Wiley, 1958).

[22] ATR Speech Database, `http://www.ctr.atr.co.jp/ATRI/Results/public/ctr/onsei.html` (Since 1987).

[23] A. Babloyantz and A. Destexhe, "Low-dimensional chaos in an instance of epilepsy," Proceedings of the National Academy of Sciences, USA **83** (1986) 3513-3517.

[24] A. Babloyantz, J. M. Salazar, and C. Nicolis, "Evidence of chaotic dynamics of brain activity during the sleep cycle," Physics Letters A **111** (1985) 152-156.

[25] P. Baldi and A. F. Atiya, "How delays affect neural dynamics and learning," IEEE Transactions on Neural Networks **5** (1994) 612-621.

[26] M. Banbrook, S. McLaughlin, and I. Mann, "Speech characterization and synthesis by nonlinear methods," IEEE Transactions on Speech & Audio Processing **7 (1)** (1999) 1-17.

[27] F. H. Brannin, Jr., "Widely convergent method for finding multiple solutions of simultaneous nonlinear equations," IBM Journal Res. Dev. (1972) 504–522.

[28] D. S. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," Physica D **20** (1986) 217-236.

[29] M. J. Bünner, M. Popp, Th. Meyer, A. Kittel, U. Rau, and J. Parisi, "Recovery of scalar time-delay systems from time series," Physics Letters A **211** (1996) 345–349.

[30] Y. J. Cao and Q. H. Wu, "A note on stability of analog neural networks with time delays," IEEE Transactions on Neural Networks **7** (1996) 1533–1535.

[31] T. L. Carroll and L. M. Pecora, "Synchronizating chaotic circuits," IEEE Transactions on Circuits & Systems **38 (4)** (1991) 453–456.

[32] M. Casdagli, "Nonlinear prediction of chaotic time series," Physicna D **35** (1989) 335–356.

[33] M. Casdagli, "Recurrence plots revisited," Physica D **108** (1997) 12–44.

[34] M. Casdagli and S. Eubank, Eds., *Nonlinear Modeling and Forecasting*, (Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, 1992).

[35] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos," *Neural Networks* **8-6** (1995) 915–930.

[36] L. Chen and K. Aihara, "Chaos and asymptotical stability in discrete-time neural networks," Physica D **104** (1997) 286–325.

[37] P. Chossat and M. Golubitsky, "Iterates of maps with symmetry," *SIAM Journal of Math. Anal.* **19** (1988) 1259–1270.

[38] P. Chossat and M. Golubitsky, "Symmetry-increasing bifurcations of chaotic attractors," *Physica D* **32** (1988) 423–436.

[39] A. C. C. Coolen and C. C. A. M. Gielen, "Delays in neural networks," Europhysics Letters **7-3** (1988) 281–285.

[40] M. C. Cross and P. C. Hohenberg, "Pattern formation outside of equilibrium," Review of Modern Physics **65 (3)** (1993) 851–1112.

[41] J. P. Crutchfield and B. S. McNamara, "Equations of motion from a data series," Complex Systems **1** (1987) 417–452.

[42] K. M. Cuomo and A. V. Oppenheim, "Circuit implementation of synchronized chaos with applications to communications," Physical Review Letters **71 (1)** (1993) 65–68.

[43] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Math. Control Signals Systems **2** (1989) 303–314.

[44] H. Dedieu, M. P. Kennedy, and M. Hasler, "Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits," IEEE Transactions on Circuits & Systems **40 (10)** (1993) 634–642.

[45] H. Degn, A. V. Holden, and L. F. Olsen, Eds., *Chaos in biological systems*, (Plenum Press, New York, 1987).

[46] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems* (Addison-Wesley, 1989).

[47] L. Dolanský and P. Tjernlund, "On certain irregularities of voiced speech waveforms," IEEE Transactions on AU **16 (1)** (1968) 51–56.

[48] K. Doya, "A study of learning algorithms for continuous-time recurrent neural networks," Ph.D. thesis, University of Tokyo (1991).

[49] K. Doya and S. Yoshizawa, "Adaptive neural oscillator using continuous-time backpropagation algorithm," Neural Networks **2** (1989) 375–385.

[50] J.-P. Eckman, S. O. Kamphorst, and D. Ruells, "Recurrence plots of dynamical systems," Europhysics Letters **4 (9)** (1987) 973–977.

[51] J. P. Eckmann, S. O. Kanmphorst, D. Ruelle, and S. Ciliberto, "Liapunov exponents from a time series," Physical Review A **34 (6)** (1986) 4971–4979.

[52] J. L. Elman and D. Zipser, "Learning the hidden structure of speech," Journal of the Acoustical Society of America **83** (1988) 1615–1626.

[53] J. D. Farmer, "Chaotic attractors of an infinite-dimensional dynamical system," Physica D **4** (1982) 366–393.

[54] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," Physical Review Letters **59** (1987) 845–848.

[55] M. J. Feigenbaum, "Quantitative universality for a class of nonlinear transformations," Journal of Statistical Physics **19** (1978) 25–52.

[56] R. FitzHugh, "Impulse and physiological states in theoretical models of nerve membrane," Biophys. Journal **1** (1961) 445–466.

[57] A. M. Fraser, "Information and entropy in strange attractors," IEEE Transactions on Information Theory **35** (1989) 245–262.

[58] A. M. Fraser and H. L. Swinney, "Independent coordinate for strange attractors from mutual information," Physical Review A **33** **(2)** (1986) 1134–1140.

[59] W. J. Freeman, "Simulation of chaotic EEG patterns with a dynamic model of the olfactory system," *Biological Cybernetics* **56** (1987) 139–150.

[60] W. J. Freeman, "Strange attractors that govern mammalian brain dynamics shown by trajectories of electroencephalographic (EEG) potential," IEEE Transactions on Circuits & Systems **35** (1988) 781–783.

[61] W. J. Freeman, "The physiology of perception," *Scientific American* **Feb** (1991) 34–41.

[62] W. J. Freeman and C. A. Skarda, "Spatial EEG patterns, nonlinear dynamics and perception: The neo-Sherringtonial view," *Brain Research Reviews* **10** (1985) 147–175.

[63] W. J. Freeman and B. W. van Dijk, "Spatial patterns of visual cortex fast EEG during conditioned reflex in a rhesus monkey," *Brain Research* **422** (1987) 267–276.

[64] W. J. Freeman, Y. Yao, and B. Burke, "Central pattern generating and recognizing in olfactory bulb: A correlation learning rule," *Neural Networks* **1** (1988) 277–288.

[65] T. Fujita, K. Yasuda, and R. Yokoyama, "Global optimization method using chaos in dissipative system," Transactions of IEICE **J77-A** **(6)** (1994) 881–889.

[66] T. Fujita, T. Watanabe, K. Yasuda, and R. Yokoyama, "Global optimization method using intermittency chaos," Transactions of IEICE **J78-A** **(11)** (1995) 1485-1493.

[67] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," Neural Networks **2** (1989) 183–192.

[68] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," Neural Networks **6** (1993) 801–806.

[69] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, San Francisco, 1979).

[70] P. Gaspard, R. Kapral, and G. Nicolis, "Bifurcation phenomena near homoclinic systems: A two-parameter analysis," Journal of Statistical Physics **35** (1986) 697–727.

[71] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," IEEE Transactions on Pattern Analysis & Machine Intelligence **6** (1984) 721–741.

[72] A. O. Girewank, "Generalized descent for global optimization," Journal of Optical Theory & Applications, **34 (1)** (1981)

[73] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning* (Addison Wesley, 1989).

[74] J. P. Gollub and S. V. Benson, "Many routes to turbulence convection," Journal of Fluid Mechanics **100** (1980) 449–470.

[75] J. P. Gollub and H. L. Swinney, "Onset of turbulence in a rotating fluid," Physical Review Letters **35** (1975) 927–930.

[76] M. Golubitsky, I. N. Stewart, and D. G. Schaeffer, *Singularities and groups in bifurcation theory II*, Applied Mathematical Science Series, **69** (Springer-Verlag, Berlin, 1988).

[77] K. Gouhara, H. Takase, Y. Uchikawa, and K. Iwata, "Attractor learning of recurrent neural networks," Artificial Neural Networks **2** (1992) 371–374.

[78] P. Grassberger and I. Procassia, "Measuring the strangeness of strange attractors," Physica D **9** (1983) 189–208.

[79] C. Grebogi, E. Ott, F. Romeiras, and J. A. York, "Critical exponents for crisis-induced intermittency," *Physical Review A* **36** (1987) 5365–5380.

[80] C. Grebogi, E. Ott, and J. A. York, "Chaotic attractors in crisis," *Physical Review Letters* **48** (1982) 1507–1510.

[81] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, New York, 1983).

[82] J. K. Hale, *Theory of Functional Differential Equations* (Springer, New York, 1977).

[83] M. Hasegawa, T. Ikeguchi, and K. Aihara, "Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems," Physical Review Letters **79 (12)** (1997) 2344–2347.

[84] M. Hata, "Euler's finite difference scheme and chaos in R^n," Proceedings of the Japan Academy, vol. 58, ser. A (1982) 178–181.

[85] D. O. Hebb, *The Organization of Behavior*, (Wiley, New York, 1949).

[86] S. U. Hegde, J. L. Sweet, and W. B. Levy, "Determination of parameters in a Hopfield/Tank computational network," *Proceeding of the IEEE International Conference on Neural Networks,* San Diego, **II** (1988) pp. 291–298.

[87] M. Held and R. M. Karp, "The traveling salesman problem and minimum spanning tree," Operations Research **18** (1970) 1138–1162.

[88] M. Hénon, "A two-dimensional mapping with a strange attractor," Communications in Mathematical Physics **50** (1976) 69–77.

[89] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, 1991).

[90] A. Herz, B. Sulzer, R. Kühn, and J. L. van Hemmen, "Hebbian learning reconsidered," Biological Cybernetics **60** (1989) 457–467.

[91] H. Herzel, D. Berry, I. R. Titze, and M. Saleh, "Analysis of vocal disorders with method from nonlinear dynamics," Journal of Speech & Hearing Research **37** (1994) 1008–1019.

[92] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proceedings of the National Academy of Sciences, USA **79** (1982) 2554–2558.

[93] J. J. Hopfield, "Neurons with graded responses have collective computational properties like those of two-state neurons," Proceedings of the National Academy of Sciences, USA **81** (1984) 3088–3092.

[94] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," Biological Cybernetics **52** (1985) 141–152.

[95] J. J. Hopfield and D. W. Tank, "Neural architecture and biophysics for sequence recognition," In Byrne & Berry Eds., *Neural Models of Plasticity* (San Diego, Academic Press, 1989) pp. 363–377.

[96] K. Hornik, "Multilayer feedforward networks are universal approximators," Neural Networks, **2** (1989) 359–366.

[97] J. L. Hudson and J. C. Mankin, "Chaos in the Belousov-Zhabotinskii reaction," Journal of Chemical Physics **74** (1981) 6171

[98] T. Ifukube, M. Hashiba, and J. Matsushima, "A role of 'waveform fluctuation' on the naturality of vowels," Journal of the Acoustical Society of Japan **47 (12)** (1991) 903–910.

[99] K. Ikeda and K. Matsumoto, "High-dimensional chaotic behavior in systems with time delayed feedback," Physica D **29** (1987) 223–235.

[100] K. Ikeda, K. Otsuka, and K. Matsumoto, "Maxwell-Bloch turbulence," *Progress of Theoretical Physics, Supplement* **99** (1989) 295–324.

[101] T. Ikeguchi, "A study of time series analysis based on nonlinear dynamical systems theory," Ph.D. thesis, Science University of Tokyo (1996).

[102] T. Ikeguchi and K. Aihara, "Estimating correlation dimensions of biological time series with a reliable method," International Journal of Fuzzy Systems **5** **(1)** (1997) 33–52.

[103] S. Ishii and M. Sato, "Chaotic potts spin model for combinatorial optimization problem," Neural Networks **10** (1997) 941–963.

[104] H. Isliker and J. Kurths, "A test for stationarity: Finding parts in a time series apt for correlation dimension estimates," International Journal of Bifurcation and Chaos **3** **(6)** (1993) 1573–1579.

[105] K. Judd, "An improved estimator of dimension and some comments on providing confidence intervals," Physica D **56** (1992) 216–228.

[106] K. Kaneko, "Pattern dynamics in spatiotemporal chaos: Pattern selection, diffusion of defect and pattern competition intermittency," Physica D **34** (1989) 1–41.

[107] K. Kaneko, "Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements," *Physica D* **41** (1990) 137-172.

[108] K. Kaneko, "Globally coupled circle maps," *Physica D* **54** (1991) 5–19.
3 **(3)** (1993) 617–623.

[109] D. T. Kaplan and L. Glass, "Coarse-grained embeddings of time series: Random walks, Gaussian random processes, and deterministic chaos," Physica D **64** (1993) 431–454.

[110] J. L. Kaplan and J. A. York, "Chaotic behaviour of multi-dimensional difference equations in functional differential equations and approximations of fixed points," In *Lecture Notes in Math.* (Springer, Berlin) **730** (1987) 204–227.

[111] M. B. Kennel, "Statistical test for dynamical nonstationarity in observed time series data,," Physical Review E **56** **(1)** (1997) 316–321.

[112] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometric construction," Physical Review A **45** **(6)** (1992) 3403–3411.

[113] M. Kerszberg and A. Zippelius, "Synchronization in neural assemblies," Physica Scripta. **T33** (1990) 54–64.

[114] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science* **220** (1983) 671-680.

[115] D. Kleinfeld, "Sequential state generation by model neural networks," Proceedings of the National Academy of Sciences, USA **83** (1986) 9469–9473.

[116] T. Kobayashi and H. Sekine, "The role of fluctuations in fundamental period for natural speech synthesis," Journal of the Acoustical Society of Japan **47** **(8)** (1991) 539–544.

[117] T. Kohda, "Indirect time series analysis of 1-d chaos and its application to designs of noise generators and to evaluations of randomness of pseudorandom number generator," In Yamaguti Ed., *Towards the Harnessing of Chaos* (Elsevier, 1994) pp. 241–256.

[118] T. Kohda, *Discrete Dynamics and Chaos*, (Corona-sha, 1998).

[119] T. Kohonen, "Correlation matrix memories," IEEE Transactions on Computers **C-21** (1972) 353–359.

[120] M. Komuro and K. Aihara, "Hierarchical structure among invariant subspaces of chaotic neural networks," preprint

[121] O. Komuro and H. Kasuya, "Characteristic of fundamental period variation and its modeling," Journal of the Acoustical Society of Japan **47** **(12)** (1991) 928–934.

[122] G. Kubin, "Nonlinear processing of speech," In Kleijin & Paliwal Eds., *Speech Coding and Synthesis* (Elsevier Sci. B. V., 1995) pp. 557–610.

[123] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: Prediction and system modeling," Technical Report of the Los Alamos National Laboratory (1987) LA-UR-87-2662.

[124] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *The Traveling Salesman Problem* (Wiley, 1985).

[125] A. V. Levy and A. Montalvo, "The tunneling algorithm for the global minimization of functions," SIAM Journal of Sci. Stat. Comput., **6** **(1)** (1981) 15–29.

[126] T.-Y. Li and J. A. York, "Period three implies chaos," American Mathematical Monthly **82** (1975) 985–992.

[127] S. Lin and B. W. Kernighan, "An efficient heuristic algorithm for the traveling salesman problem," Operations Research **21** (1973) 498–516.

[128] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communication bf COM-28 (1) (1980) 84–95.

[129] F. Lombard and J. D. Hart, "The analysis of change point data with dependent errors," In Carlstein, Muller, and Siegmund Eds., *Change Points Problems*, IMS Lecture Notes - Monograph Series, **23**, (Inst. Math. Statist., Hayward, CA, 1994) pp. 194–209.

[130] E. N. Lorenz, "Deterministic nonperiodic flow," Journal of Atmos. Sci. **20** (1963) 901–912.

[131] M. Lu and K. Shimizu, "Global optimization of function of several variables by ε-approximation approach," Transactions of IEICE **J75-A** **(4)** (1992) 769–777.

[132] D. G. Luenberger, *Linear and Nonlinear Programming* (Addison-Wesley, 1973).

[133] R. Manuca and R. Savit, "Stationarity and nonstationarity in time series analysis," Physica D **99** (1996) 134–161.

[134] G. Matsumoto, K. Aihara, Y. Hanyu, N. Takahashi, S. Yoshizawa, and J. Nagumo, "Chaos and phase locking in normal squid axons," Physics Letters A **123** (1987) 162-166.

[135] T. Matsumoto, M. Komuro, H. Kokubu, and R. Tokunaga, *Bifurcations: Sights, Sounds and Mathematics*, (Springer-Verlag, 1993).

[136] R. M. May, "Simple mathematical models with very complicated dynamics," Nature **261** (1976) 459–467.

[137] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," Science **197** (1977) 287–295.

[138] C. M. Marcus and R. M. Westervelt, "Stability of analog neural networks with delay," Physical Review A **39** **(2)** (1989) 347–359.

[139] A. I. Mees, "Dynamical systems and tesselations: Detecting determinism in data," International Journal of Bifurcation and Chaos **1** **(4)** 777–794.

[140] A. I. Mees, P. E. Rapp, and L. S. Jennings, "Singular-value decomposition and embedding dimension," Physical Review A **36** **(1)** (1987) 341–346.

[141] W. Mende, H. Herzel, and I. R. Titze, "Bifurcations and chaos in newborn cries," Physics Letters A **145** (1990) 418–424.

[142] T. Miyano, "Are Japanese vowels chaotic?," Proceedings of the 4th International Conference on Soft Computing **2** (1996) pp. 634–637.

[143] T. Miyano, A. Nagami, I. Tokuda, and K. Aihara, "Detecting nonlinear dynamics in voiced sound of Japanese vowel /a/ by time series prediction," International Journal of Bifurcation and Chaos, *in press.*

[144] T. Miyano, S. Kimoto, H. Shibuta, K. Nakashima, Y. Ikenaga, and K. Aihara, "Time series analysis and prediction on complex dynamical behavior observed in a blast furnace," Physica D **135** (2000) 305–330.

[145] M. C. Mozer, "A focused backpropagation algorithm for temporal pattern recognition," Complex Systems **3** (1989) 349–381.

[146] S. Murashige and K. Aihara, "Experimental study on chaotic motion of a flooded ship in waves," Proceedings of the Royal Society of London **454** (1998) 2537–2553.

[147] T. Nagashima, J. Miyazaki, Y. Shiroki, and I. Tokuda, "Synchronization and its dynamic recombination in associative memory based on a chaotic neural network," International Journal of Chaos Theory and Applications **2 (2)** (1997) 1–24.

[148] J. Nagumo, S. Arimoto, S. Yoshizawa, "An active pulse transmisson line simulating nerve axon," Proc. Inst. Radio Eng. **50** (1962) 2061–2070.

[149] K. Nakano, "Association: A model of associative memory," IEEE Transactions on Systems, Man, and Cybernetics **SMC-2** (1972) 380–388.

[150] S. Nara, P. Davis, M. Kawachi, and H. Totsuji, "Chaotic memory dynamics in a recurrent neural network with cycle memories embedded by pseudo-inverse method," International Journal of Bifurcation and Chaos **5 (4)** (1995) 1205–1212.

[151] S. Nara, P. Davis, and H. Totsuji, "Memory search using complex dynamics in a recurrent neural network model," Neural Networks **6** (1993) 963–973.

[152] S. S. Narayanan and A. A. Alwan, "A nonlinear dynamical systems analysis of fricative consonants," Journal of the Acoustical Society of America **97 (4)** (1995) 2511–2524.

[153] G. Nicolis and I. Prigogine, *Self-Organization in Nonequilibrium Systems*, (Wiley, 1977).

[154] H. Nozawa, "A neural network model as a globally coupled map and applications based on chaos," *Chaos* **2-3** (1992) 377-386.

[155] E. Ott, C. Grebogi, and J. A. York, "Controlling chaos," Physical Review Letters **64** (1990) 1196–1199.

(1980) 712–716.

[156] K. Pakdaman and C. P. Malta, "A note on convergence under dynamical threshold with delays," IEEE Transactions on Neural Networks **9 (1)** (1998) 231–233.

[157] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems* (Springer-Verlag, 1989).

[158] B. A. Pearlmuter, "Learning state space trajectories in recurrent neural networks," Neural Computation **1** (1989) 263–269.

[159] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," Physical Review Letters **64 (8)** (1990) 821–824.

[160] M. M. Peixoto, "Structural stability in the plane with enlarged boundary conditions," Topology **1** (1962) 101–120.

[161] G. Pérez and H. A. Cerdeira, "Extracting messages masked by chaos," Physical Review Letters **74 (11)** (1995) 19700–1973.

[162] F. J. Pineda, "Dynamics and architecture for neural computation," *Journal of Complexity* 4 (1988) 216–245.

[163] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks," Physical Review Letters **59** (1987) 2229–2232.

[164] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Optimization Problems* (Blackwell, Oxford, 1993).

[165] O. E. Rössler, "Continuous chaos," Annals of the New York Academy of Science **31** (1979) 376–392.

[166] J.-C. Roux, "Experimental studies of bifurcations leading to chaos in the Belousov-Zhabotinskii reactions," Physica D **7** (1983) 57–68.

[167] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature **323** (1986) 533–536.

[168] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing* (MIT Press, Cambridge, 1986).

[169] Y. Saida, T. Matsuno, and K. Yamafuji, Sensors and Materials **4** (1992) 135

[170] M. Sano and Y. Sawada, In Tatsumi Ed., *Trubulence and Chaotic Phenomena in Fluids* (North-Holland, 1984) p. 167

[171] M. Sano and Y. Sawada, "Measurement of the Lyapunov spectrum from chaotic time series," Physical Review Letters **55** (1985) 1082–1085.

[172] M. Sato, "A learning algorithm to teach spatio-temporal patterns to recurrent neural networks," Biological Cybernetics **62** (1990) 259–263.

[173] M. Sato, "A real time learning algorithm for recurrent analog neural networks," Biological Cybernetics **62** (1990) 237–241

[174] M. Sato, K. Joe, and T. Hirahara, "APOLONN brings us to the real world," Proceedings of the International Joint Conference of Neural Networks **1** (1990) 581–587

[175] M. Sato, Y. Murakami, and K. Joe, "Learning chaotic dynamics by recurrent neural networks," Proceedings of the International Conference on Fuzzy Logic & Neural Networks (1990) 601–604.

[176] T. Sauer, J. A. York, and M. Casdagli, "Embedology," Journal of Statistial Physics **65** **(3)** (1991) 579–616.

[177] T. Schreiber, "Detecting and analyzing nonstationarity in a time series using nonlinear cross predictions," Physical Review Letters **78** **(5)** (1997) 843–846.

[178] T. Schreiber and A. Schmitz, "Classification of time series data with nonlinear similarity measures," Physical Review Letters **79** **(8)** (1997) 1475–1478.

[179] L. P. Shil'nikov, "A case of the existence of a countable number of periodic motions," Sov. Math. Dokl. **6** (1965) 163–166.

[180] L. A. Smith, K. Godfrey, P. Fox, and K. Warwick, "A new technique for fault detection in multi-sensor probes," In *Control 91*, **1**, (IEE Conference Publication 332, 1991) pp. 1062–1067.

[181] H. Sugata and K. Shimizu, "Global optimization using chaos in a quasi-steepest descent method," Transactions of IEICE **J79-A** **(3)** (1996) 658–668.

[182] G. Sugihara and R.M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series," Nature **344** (1990) 734–741.

[183] C. A. Skarda and W. J. Freeman, "How brains make chaos in order to make sense of the world," Behavioral and Brain Sciences **10** (1987) 161–195.

[184] H. Sompolinsky and I. Kanter, "Temporal association in asymmetric neural networks," Physical Review Letters **57** (1986) 2861–2864.

[185] T. Takanami and G. Kitagawa, "Estimation of arrival times of seismic waves by multivariate time series models," Ann. Inst. Statist. Math. **43** **(3)** (1991) 407–433.

[186] F. Takens, "Detecting strange attractors in turbulence," In *Lecture Notes in Math.* **898** (Springer, Berlin, 1981) pp. 366–381.

[187] J. Tani, "Proposal of chaotic steepest descent method for neural networks and analysis of their dynamics," Transactions of IEICE **J74-A** **(8)** (1991) 1208–1215.

[188] D. W. Tank and J. J. Hopfield, "Neural computation by concentrating information in time," Proceedings of the National Academy of Sciences, USA **84** (1987) 1896–1900.

[189] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: The method of surrogate data," Physica D **58** (1992) 77–94.

[190] J. M. T. Thompson and H. B. Stewart, *Nonlinear Dynamics and Chaos: Geometrical Methods for Engineers and Scientists*, (John Wiley, 1986).

[191] I. R. Titze, R. J. Baken, H. Herzel, "Evidence of chaos in vocal fold vibration," In Titze Ed., *Vocal Fold Physiology* (Singular, San Diego, 1993), pp. 143–188.

[192] I. Tokuda, "Approximation capability of neural networks with time-delayed feedbacks," Technical Report of IEICE (1998) **NC97-75, NLP97-123**.

[193] I. Tokuda and K. Aihara, "Back-propagation learning of infinite-dimensional dynamical systems," Technical Report of IEICE (1997) **NC96-92, NLP96-138**.

[194] I. Tokuda, K. Aihara, R. Tokunaga, T. Nagashima, "Learning algorithm for chaotic neural networks which solve optimization problems," Proceedings of The Symposium on Nonlinear Theory and its Applications (Hawaii) **1** (1997) pp. 617–620.

[195] I. Tokuda, Y. Hirai, and R. Tokunaga, "Back-propagation learning of an infinite-dimensional dynamical system," Proceedings of the International Joint Conference of Neural Networks (Nagoya) **3** (1993) pp. 2271–2275.

[196] I. Tokuda, R. Tokunaga, and K. Aihara, "Laryngeal vibrations: Towards understanding causes of irregularities in speech signals of vowel sounds," In Yamaguti Ed., *Towards the Harnessing of Chaos* (Elsevier, Amsterdam, 1994) pp. 413–418.

[197] I. Tokuda, R. Tokunaga, and K. Aihara, "A simple geometrical structure underlying speech signals of the Japanese Vowel /a/," International Journal of Bifurcation and Chaos **6** (1996) 149–160.

[198] R. Tokunaga, S. Kajiwara, and T. Matsumoto, "Reconstructing bifurcation diagrams only from time waveforms," Physica D **79** (1994) 348–360.

[199] B. Townshend, "Nonlinear prediction of speech signals," In Casdagli & Eubank Eds., *Nonlinear Modeling and Forecasting*, (SFI Studies in Sciences of Complexity, Addison-Wesley, 1992) pp. 433–453.

[200] I. Tsuda, "Chaotic itinerancy as a dynamical basis of Hermeneutics in brain and mind," *World Futures* **32** (1991) 167-184.

[201] I. Tsuda, "Chaotic neural networks and thesaurus," In Holden and Kryukov Eds., *Neurocomputers and Attention I,* (Manchester University Press, 1991) pp. 405-424.

[202] I. Tsuda, "Dynamics link of memory: Chaotic memory map in nonequilibrium neural networks," *Neural Networks* **5** (1992) 313-326.

[203] I. Tsuda, "Can stochastic renewal of maps be a model for cerebral cortex?," Physica D **75** (1994) 165-178.

[204] I. Tsuda, E. Koerner, and H. Shimizu, "Memory dynamics in asynchronous neural networks," *Progress of Theoretical Physics* **78** (1987) 51-71.

[205] I. Tsuda, T. Tahara, and H. Iwanaga, "Chaotic pulsation in human capillary vessels and its dependence on mental and physical conditions," International Journal of Bifurcation and Chaos **2 (2)** (1994) 313–324.

[206] H. Voss and J. Kurths, "Reconstruction of non-linear time delay models from data by the use of optimal transformations," Physics Letters A **234** (1997) 336–344.

[207] M. Watanabe, K. Aihara, S. Kondo, "Automatic Learning in Chaotic Neural Networks," Transactions of IEICE **J78-A (6)** (1995) 686–691.

[208] R. Wayland, D. Bromely, D. Pickett, and A. Passamante, "Recognizing determinism in a time series," Physical Review Letters **70 (5)** (1993) 580–582.

[209] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," Neural Computation **1** (1989) 270–280.

[210] G. V. Willson and G. S. Pawley, "On the stability of the travelling salesman problem algorithm of Hopfield and Tank," *Biological Cybernetics* **58** (1988) 63–70.

[211] A. Witt, J. Kurths, and A Pikovsky, "Testing stationarity in time series," Physical Review E **58** (1998) 1800–1810.

[212] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," Physica D **16** (1985) 285–317.

[213] T. Yamada and K. Aihara, "Dynamical neural networks and the traveling salesman problem," In Yamaguti Ed., *Towards the Harnessing of Chaos* (Elsevier, Amsterdam, 1994) pp. 423-426.

[214] M. Yamaguti, Ed., *Towards the Harnessing of Chaos* (Elsevier, Amsterdam, 1994).

[215] M. Yamaguti and H. Matano, "Euler's finite difference scheme and chaos," Proceedings of the Japan Academy, ser. A, no. 55 (1979) 78–80.

[216] Y. Yao and W. J. Freeman, "Models of biological pattern recognition with spatially chaotic dynamics," *Neural Networks* **3** (1987) 153–170.

[217] T. Yoshinaga, Y. Inuma, and H. Kawakami, "Analysis of chaotic itinerancy observed in a chaotic neural network," Technical Report of IEICE (1997) **NLP97-102.**

[218] S. Yoshizawa, H. Osada, J. Nagumo, "Pulse sequences generated by a degenerate analog neuron model," Biological Cybernetic **45** (1982) 23

Appendix A

Numerical technique and mathematical preliminaries

A-1 Numerical technique

The procedure for minimizing the cost function (3.12) of Chapter 3 via the quasi-Newton method requires calculating of the cost function $U(\Omega)$ and its first derivatives $\partial U/\partial\Omega$. This appendix illustrates practical numerical algorithms for computing $U(\Omega)$ and $\partial U/\partial\Omega$.

Let us approximated the solution curve $\phi^t : R^L \times R^d \to R^d$ generated form the nonlinear predictor (3.8) with an initial condition $\phi^0(\Omega, X) = X$ by using the Euler's integration formula with a time step of $\Delta t/q$ $(q \in Z)$ as

$$\begin{aligned} \phi^{\frac{\Delta t}{q}}(\Omega, X) &= X + \frac{\Delta t}{q}\,\alpha\,F(\Omega, X) & \text{(A-1-1)} \\ &\overset{\text{def}}{=} G(\Omega, X). & \text{(A-1-2)} \end{aligned}$$

Then, the cost function of eq. (3.12) can be rewritten as

$$U(\Omega) = \sum_{j=(d-1)\tau+1}^{N-W-K} \sum_{k=1}^{K} \frac{1}{2} \mid G^{kq}(\Omega, X_j) - X_{j+k} \mid^2 \qquad \text{(A-1-3)}$$

where G^{kq} is calculated by the iterative procedure of

$$G^{n+1}(\Omega, X) = G(\Omega, G^n(\Omega, X)). \qquad \text{(A-1-4)}$$

With respect to the cost function (A-1-3), the first derivatives $\partial U/\partial\Omega$ are derived as

$$\frac{\partial U}{\partial\Omega}(\Omega) = \sum_{j=(d-1)\tau+1}^{N-W-K} \sum_{k=1}^{K} {}^T(G^{kq}(\Omega, X_j) - X_{j+k}) \cdot \frac{\partial G^{kq}}{\partial\Omega}(\Omega, X_j) \qquad \text{(A-1-5)}$$

where the right hand values $\partial G^{kq}/\partial\Omega$ are calculated by solving the first variational equations of (A-1-4)

$$\frac{\partial G^{n+1}}{\partial\Omega}(\Omega, X) = \frac{\partial G}{\partial\Omega}(\Omega, G^n(\Omega, X)) + \frac{\partial G}{\partial X}(\Omega, G^n(\Omega, X))\frac{\partial G^n}{\partial\Omega}(\Omega, X) \qquad \text{(A-1-6)}$$

with an initial condition $\partial G^0/\partial\Omega = 0$ [209].

In our experiment, we set $q = 4$ and integrated the differential equation (3.8) with a time step of $\Delta t = 0.05$.

A-2 Symmetry in chaotic neural network

This appendix provides mathematical proof of the symmetry in chaotic neural networks studied in Chapter 5. We show (i): $f \circ \gamma = \gamma \circ f$ and (ii): $f \circ \eta = \eta \circ f$. (i) and (ii) immediately lead to $f \circ (\eta^l \circ \gamma^m) = (\eta^l \circ \gamma^m) \circ f$ for $l = 0, 1$ and $m = 0, .., N-1$.

(i) For any $p \in R^{N \times N}$, $f \circ \gamma(p) = \gamma \circ f(p)$ because

$$\begin{aligned}
\{f \circ \gamma(p)\}_{ik} &= r\, p_{i\,k+1} + (1-r)\, \sigma\, (\sum_{j=0}^{N-1} \sum_{l=0}^{N-1} T_{ik,\,jl}\, p_{j\,l+1} + I) && \text{(A-2-1)}\\
&= r\, p_{i\,k+1} + (1-r)\, \sigma\, (\sum_{j=0}^{N-1} sum_{l'=0}^{N-1} T_{ik,\,j\,l'-1}\, p_{jl'} + I) && \text{(A-2-2)}\\
&= r\, p_{i\,k+1} + (1-r)\, \sigma\, (\sum_{j=0}^{N-1} \sum_{l'=0}^{N-1} T_{i\,k+1,\,jl'}\, p_{jl'} + I) && \text{(A-2-3)}\\
&= \{f(p)\}_{i\,k+1} = \{\gamma \circ f(p)\}_{ik} && \text{(A-2-4)}
\end{aligned}$$

for $0 \leq\ i, k\ \leq N-1$.

The indices in $\{I_{ik}\}$ are dropped since the set of the parameters $\{I_{ik}\}$ takes a same value by eq. (5.9). From eq. (A-2-2) to eq. (A-2-3), we have used $T_{ik,\,j\,l-1} = T_{i\,k+1,\,jl}$, which is the property of the synaptic connections defined by eqs. (5.7)-(5.8).

(ii) For any $p \in R^{N \times N}$, $f \circ \eta = \eta \circ f$ because

$$\begin{aligned}
\{f \circ \eta(p)\}_{ik} &= r\, p_{i\,N-k} + (1-r)\, \sigma\, (\sum_{j=0}^{N-1} \sum_{l=0}^{N-1} T_{ik,\,jl}\, p_{j\,N-l} + I) && \text{(A-2-5)}\\
&= r\, p_{i\,N-k} + (1-r)\, \sigma\, (\sum_{j=0}^{N-1} \sum_{l'=0}^{N-1} T_{ik,\,j\,N-l'}\, p_{jl'} + I) && \text{(A-2-6)}\\
&= r\, p_{i\,N-k} + (1-r)\, \sigma\, (\sum_{j=0}^{N-1} \sum_{l'=0}^{N-1} T_{i\,N-k,\,jl'}\, p_{jl'} + I) && \text{(A-2-7)}\\
&= \{f(p)\}_{i\,N-k} = \{\eta \circ f(p)\}_{ik} && \text{(A-2-8)}
\end{aligned}$$

for $0 \leq\ i, k\ \leq N-1$.

From eq. (A-2-6) to eq. (A-2-7), we have used $T_{ik,\,j\,N-l} = T_{i\,N-k,\,jl}$, which is the property of the synaptic connections defined by eqs. (5.7)-(5.8).

A-3 Conjugate attractors with equivalent TSP solution

This appendix shows how the system symmetry gives rise to $2N$ conjugate attractors representing an equivalent TSP solution in Chapter 5.

Suppose there exists an asymmetric attractor $O = \{p(n) \mid n = 0, 1, ..\}$ which is coded as a feasible TSP solution $\bar{J}(\bar{p})$, where $\bar{p} = \lim_{T\to\infty}(1/T)\sum_{n=0}^{T-1} p(n)$ represents a long-term averaged firing rate of O.

For all $l = 0, 1$ and $m = 0, .., N-1$, a long-term averaged firing rate of a conjugate attractor $\eta^l \circ \gamma^m(O)$ $(= \{\eta^l \circ \gamma^m((p(n)) \mid n = 0, 1, ..\})$ can be written as $\eta^l \circ \gamma^m(\bar{p})$ because a linear mapping $\eta^l \circ \gamma^m$ transforms $\bar{p}$ into $\eta^l \circ \gamma^m(\bar{p}) = \lim_{T\to\infty}(1/T)\sum_{n=0}^{T-1} \eta^l \circ \gamma^m(p(n))$. Since $\bar{J}(\eta^l \circ \gamma^m(\bar{p})) = \eta^l \circ \gamma^m(\bar{J}(\bar{p}))$, the $2N$ conjugate asymmetric attractors $\{\eta^l \circ \gamma^m(O) \mid l = 0, 1,\ m = 0, .., N-1\}$ are coded respectively as $\{\eta^l \circ \gamma^m(\bar{J}(\bar{p})) \mid l = 0, 1,\ m = 0, .., N-1\}$.

It is well known that the $2N$ codes $\{\eta^l \circ \gamma^m(\bar{J}(\bar{p})) \mid l = 0, 1,\ m = 0, .., N-1\}$ represent an equivalent TSP solution [94]. This is because the transformations γ and η do not change the basic tour configuration. Namely, γ only shifts a choice of an initial city in a visiting order of the cities and η changes a choice of a visiting direction of the cities. Hence, the $2N$ conjugate attractors $\{\eta^l \circ \gamma^m(O) \mid l = 0, 1,\ m = 0, .., N-1\}$ are coded into an equivalent TSP solution.

Appendix B

主論文を構成する論文

B-1 原著論文

1. Isao Tokuda, Shihoko Kajiwara, Ryuji Tokunaga, Takashi Matsumoto: "Recognizing Chaotic Time-Waveforms in terms of a Parametrized Family of Nonlinear Predictors," Physica D, Vol. 95, pp. 380-395 (1996).

2. 梶原志保子, 徳田功, 徳永隆治, 松本隆: "非線形予測子パラメータ族によるカオス的時系列データの認識," 電子情報通信学会論文誌 A, Vol. J79-A, No. 8, pp.1394-1403 (1996).

3. Isao Tokuda, Ryuji Tokunaga, Kazuyuki Aihara: "A Simple Geometrical Structure Underlying Speech Signals of the Japanese Vowel /a/," International Journal of Bifurcation and Chaos, Vol. 6, No. 1, pp. 149-160 (1996).

4. Isao Tokuda, Tomosama Nagashima, Kazuyuki Aihara: "Global Bifurcation Structure of Chaotic Neural Networks and its Application to Traveling Salesman Problems," Neural Networks, Vol. 10, No. 9, pp.1673-1690 (1997).

5. 徳田功, 小野寺浩二, 徳永隆治, 合原一幸, 長島知正: "最適化問題を解くカオス的力学系の大域的分岐のシナリオとその最適化手法の検証," 電子情報通信学会論文誌 A, Vol. J80-A, No. 6, pp.936-948 (1997).

6. 徳田功, 田村亜紀, 徳永隆治, 合原一幸, 長島知正: "最適化問題を解くカオス的力学系の学習," 電子情報通信学会論文誌 A, Vol. J81-A, No. 3, pp.377-388 (1998).

7. Isao Tokuda, Kazuyuki Aihara, Tomomasa Nagashima: "Adaptive Annealing for Chaotic Optimization," Physical Review E, Vol. 58, No. 4, pp.5157-516 (1998).

8. Isao Tokuda, Ryuji Tokunaga, Takashi Matsumoto: "Detecting Switch Dynamics in Chaotic Time-Waveform using a Parametrized Family of Nonlinear Predictors," Physica D, Vol. 135, pp.63-78 (2000).

9. Isao Tokuda, Ryuji Tokunaga, Kazuyuki Aihara: "Back-Propagation Learning of Infinite-Dimensional Dynamnical Systems," Neural Networks, *in revision.*

B-2 国際会議等論文(レフェリー付)

1. Isao Tokuda, Yuzo Hirai, Ryuji Tokunaga: "Back-propagation Learning of an Infinite-Dimensional Dynamical System," Proceedings of The International Joint Conference on Neural Networks (Nagoya) vol.3/3 pp2271-2275 (1993).

2. Shihoko Kajiwara, Isao Tokuda, Ryuji Tokunaga, Takashi Matsumoto: "Reconstructing Bifurcation Diagrams only from Chaotic Time-waveforms," Proceedings of The 3rd International Conference on Fuzzy Logic, Neural Nets, and Soft Computing (Iizuka) pp.515-517 (1994).

3. Ryuji Tokunaga, Isao Tokuda, Shihoko Kajiwara, Takashi Matsumoto: "Reconstructing Bifurcation Diagrams only from Chaotic Time-waveforms," Proceedings of The International Conference on Dynamical Systems and Chaos (Tokyo) vol.2/2 pp.345-354 (1994).

4. Isao Tokuda, Tomomasa Nagashima, Yoshitaka Shiroki, Noriyuki Kuramae: "Statistical Properties of Chaotic Neural Network as an Optimum Solution of Traveling Salesman Problem," Proceedings of The Symposium on Nonlinear Theory and its Applications (Ibusuki) pp.161-164 (1994).

5. Isao Tokuda, Tomomasa Nagashima, Kazuyuki Aihara: "Global Bifurcation Structure of Chaotic Neural Networks and its Application to Optimization Problems," Proceedings of The International Workshop on Soft Computing in Industry (Muroran) pp.287-292 (1996).

6. Isao Tokuda, Kazuyuki Aihara, Tomomasa Nagashima: "Adaptive Annealing for Chaotic Optimization," Proceedings of The Symposium on Nonlinear Theory and its Applications (Katsurahama) pp.281-284 (1996).

7. Isao Tokuda, Kazuyuki Aihara, Ryuji Tokunaga, Tomomasa Nagashima: "Learning algorithm for Chaotic Neural Networks which Solve Optimization Problems," Proceedings of The Symposium on Nonlinear Theory and its Applications (Hawaii) vol. 1/2 pp.617-620 (1997).

B-3 参考論文

1. 徳田功: "音声とカオス," 数理科学 3 月号 (サイエンス社) pp.53-59 (1995).

2. 徳田功, 合原一幸: “複雑系への挑戦 - 複雑な問題を複雑な系で解く,” KDD Technical Journal **25** 夏号 pp.8-11 (1996).

3. 徳永隆治, 松本隆, 徳田功: “分岐図再構成 - 時系列信号からの係数族推定,” 応用数理 7 巻 4 号 pp.271-282 (1997).

4. 徳田功, 合原一幸: “複雑系とカオス,” 電気学会誌 Vol. 118, No. 5, pp.294-297 (1998).

5. Isao Tokuda, Kazuyuki Aihara: “Back-Propagation Learning of Infinite-Dimensional Dynamical Systems,” 電子情報通信学会技術報告 NLP96-138, NC96-92, pp.109-114 (1997).

6. Isao Tokuda: “Approximation Capability if Neural Networks with Time-Delayed Feedbacks,” 電子情報通信学会技術報告 NLP97-123, NC97-75, pp.1-5 (1998).

7. Isao Tokuda, Ryuji Tokunaga, Kazuyuki Aihara: “Laryngeal Vibrations: Towards Understanding Causes of Irregularities in Speech Signals of Vowel Sounds,” in *Towards the Harnessing of Chaos* (M. Yamaguti Ed., Elsevier, 1994) pp.413-418

8. Isao Tokuda, Kazuyuki Aihara: “Laryngeal Vibrations as an Underlying Cause of Irregularities in Speech Signals of Vowel Sounds,” in *Nonlinear Dynamics: New Theoretical and Applied Results* (J. Awrejcewicz Ed., Akademie Verlag Berlin, 1995) pp.393-403

9. Tomomasa Nagashima, Jun Miyazaki, Yoshitaka Shiroki, Isao Tokuda: “Synchronization and its Dynamic Recombination in Associative Memory based on a Chaotic Neural Network,” International Journal of Chaos Theory and Applications, Vol. 2, No. 2, pp. 1-24 (1997).

10. Isao Tokuda, Koji Onodera, Ryuji Tokunaga, Kazuyuki Aihara, Tomomasa Nagashima: “Global Bifurcation Scenario for Chaotic Dynamical Systems that Solve Optimization Problems and Analysis of their Optimization Capability,” Electronics and Communications in Japan, Part 3, Vol. 81, No. 2, pp.1-12 (1998).

11. Shihoko Kajiwara, Isao Tokuda, Ryuji Tokunaga, Takashi Matsumoto: “Recognition of Chaotic Time-Series using a Parametrized Family of Nonlinear Predictors,” Electronics and Communications in Japan: Part 3, Vol. 81, No. 3, pp. 35-46 (1998).

12. Isao Tokuda, Aki Tamura, Ryuji Tokunaga, Kazuyuki Aihara, Tomomasa Nagashima: “Learning Algorithm for Chaotic Dynamical Systems that Solve Optimization Problems,” Electronics and Communications in Japan, Part 3, Vol. 82, No. 3, pp.10-21 (1999).

13. T. Miyano, A. Nagami, I. Tokuda, K. Aihara: "Detecting Nonlinear Dynamics in Voiced Sound of Japanese Vowel /a/ by Time Series Prediction," International Journal of Bifurcation and Chaos, *in press.*

14. I. Tokuda, T. Miyano, K. Aihara: "Surrogate Analysis for Detecting Nonlinear Dynamics in Japanese Vowels," Technical Report of The University of Tokyo, METR 99-05, August 1999; Journal of the Acoustical Society of America, *in revision.*

Appendix C

謝辞

本研究を進めるに当たり, 常日頃より暖かい御指導と御支援を頂く, 東京大学工学部計数工学科 合原一幸教授に感謝いたします。 合原先生には, 筆者の学生時代から現在に至るまで、一貫して研究の方向性を示して頂きました。 特に, 工学分野におけるカオス研究の醍醐味と脳研究の重要性を御教示いただき, 本論文における筆者の研究をお導きいただきました。 心より感謝いたします。 また, 本論文をまとめるに際し, 東京大学工学部 中村仁彦助教授, 堀田武彦助教授, 松井知己助教授, 村重淳助教授には, 貴重な御討論と御助言を賜わりました。 謹んで感謝いたします。

室蘭工業大学工学部情報工学科 長島知正教授には, 筆者が室蘭工業大学で本研究を進めるに当たって, 日頃より多大なる御支援を頂きました。 深く感謝いたします。

早稲田大学理工学部電気工学科 松本隆教授には, 共同研究を通して, 切口を変えた物の見方の面白さと率直な議論の楽しさを学びました。 深く感謝いたします。

筑波大学電子情報工学系 徳永隆治助教授には, カオス研究の創造性と研究生活の豊かさを教わり, 筆者を研究者の道にお導きいただきました。 筆者の学生時代から現在に至るまで, 公私にわたって様々の御指導と御支援をいただき, 本論文の研究について御協力いただきました。 心より感謝いたします。

脳神経系におけるカオス研究について様々の御教示をいただいた津田一郎教授, カオス的時系列解析に関して様々のアドバイスをいただいた Alistair I. Mees 教授, 音声系のカオス振動について研究討論をいただいた Hanspter Herzel 教授, 遅延を含むカオス時系列の解析手法について研究討論をいただいた Juergen Kurths 教授と Henning Voss 博士, 高次元力学系におけるカオス的遍歴現象について数理の切口をお教えいただいた小室元政助教授, 分岐理論の専門的立場からアドバイスをいただいた国府寛治助教授, 音声の非線解析について共同研究を行ない, 活発な議論をさせていただいた宮野尚哉助教授, カオス的時系列解析の様々の話題について御教授いただいた池口徹博士, 音声研究の専門的立場から様々のアドバイスをいただいた三木信弘助教授, カオスセミナーから引き続きダイナミカルブレイニクスにおいて熱心に研究討論に御参加いただいた安達雅春助教授, 市瀬夏洋博士, 山田泰司氏およびその他のセミナー参加者に感謝します。

筆者の勤務する室蘭工業大学工学部情報工学科の諸先生方と職員の皆様には, 日頃より暖かい御支援をいただき, 筆者が研究に専念できる環境に御配慮をいただいきました。 深く感謝いたします。