Chaotic Prediction and Modeling of Sea Clutter using Neural Networks

by

Henry Leung

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the degree
Doctor of Philosophy

McMaster University
March 1991

Chaotic Prediction and Modeling of Sea Clutter using Neural Networks

Doctor of Philosophy (1991)
(Electrical Engineering)

McMaster University
Hamilton, Canada

TITLE: CHAOTIC PREDICTION AND MODELING OF SEA CLUTTER USING NEURAL NETWORKS

AUTHOR: Henry Kwai Yi Leung
B.Math (University of Waterloo,1984)
M.Sc (University of Toronto,1985)
M.Eng (McMaster University,1986)

SUPERVISOR: Dr.S. Haykin

NUMBER OF PAGES: x, 182

ABSTRACT

Radar clutter, the unwanted radar echoes, has a long history of being modeled as a stochastic process. The main reason for using this model is that radar clutter appears to be very random to our naked eyes. Due to this stochastic assumption, radar detection is based on statistical decision theory. More precisely, the probability distribution of noise or clutter is obtained to derive the likelihood function for making decision.

In this thesis, we try to justify the stochastic assumption of radar clutter, in particular, sea clutter. We find that assuming sea clutter as a random process uses unnecessarily high degrees of freedom. In other words, sea clutter does not have to be modeled by a random process to handle its random behavior. Using different real-life sea clutter data, we show that the random nature of sea clutter is possibly a result of the chaotic phenomenon.

After showing that sea clutter is not truely random, we then try to model sea clutter data by a deterministic dynamical system. To construct a useful model for sea clutter, we need to reconstruct the dynamics of sea clutter, and neural network is used here as a tool to achieve this purpose. Two novel neural networks are developed to reconstruct the clutter dynamics. The first one is called rational function neural network which has an unique local minimum and a rapid learning phase. The second network, which uses the idea that sea clutter can be embedded as a manifold, does not require any learning, and is very robust and accurate. Both networks have excellent performances in reconstructing the dynamics of the real-life sea clutter data.

The model for sea clutter is then used for detection of small targets in ocean environment. Now detection is no longer a statistic decision problem, but rather a process of distinguishing two different dynamical systems. One dynamical system

contains trajectories for sea clutter and targets, and the other describes the motion of sea clutter only. We use the trajectory matching idea to classify different dynamical systems, and the result of detecting real-life small targets such as a waverider is very exciting.

ACKNOWLEDGEMENTS

I would like to express my appreciation to my supervisor, Dr.S. Haykin, for his efforts throughtout this work. I am grateful to the members of my thesis committee, Drs.J.P. Reilly, G. Wolkowicz, and P. Yip, for their contributions of time and useful suggestion. I would also like to thank my parents, sisters, brother, brother-in-law, and other well wishers who have contributed to my successes over the years through their confidence and belief in me. Most of all I thank my Lord and Saviour Jesus Christ for being with me all the way.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Sea Clutter Modeling

Radar clutter consists of radar returns from reflectors that are not of interest, and often obscures the signals from targets that are of interest. Radar clutter is typically caused by objects such as rain, birds, sea, woods, and mountains. Examples of radar targets are ships, aircraft, and satellites. However, there are also cases in which the clutter itself is the wanted target. One example is the clutter classification problem in air traffic control radar.

The radar backscatter from the sea surface is called sea clutter. A description of sea clutter is of interest to radar designers and operators, because shipboard radars viewing the ocean surface or low-flying targets must often operate at such low elevation angles that the sea surface is illuminated along with the target.

Early work by Goldstein [16] characterized sea clutter as noise-like and the average radar cross section per unit area of illuminated sea surface, and an associated distribution function to describe its amplitude characteristic. Because of the central limit theorem and simple processing architectures, this probability was conventionally

considered to be Gaussian. This follows the usual assumption that a large number of independent scatters are illuminated by the radar. However, measurements of sea clutter for very narrow pulse widths have demonstrated that a Gaussian model is inadequate. Instead, there is an increasing tendency toward high-amplitude spikes (commonly known as sea spikes) with the result that the tail of the distribution function is higher than that given by a Gaussian model.

Since then, there have been many subsequent attempts to relate the behavior of sea surface to the statistical properties of the scattered radar waves. Some progress was made in the late 1960s, the fluctuations properties of the radar return have generally been modeled by ad hoc distributions that give a reasonable fit to the data; lognormal and Weibull [36] are two examples. During the mid-seventies, the random-walk model was added to the list. A semi-empirical model, K-distribution, was proposed [24]. The fitting of this distribution to experimental data was very good, and it is still generally accepted to be the best model for sea clutter so far.

Before going any further on the modeling of sea clutter, we think we should explain what a mathematical model is, and how we choose a good model. The complexity of nature calls for partial descriptions that are sometimes mathematical in character and then generally called mathematical models. The relation between nature and a mathematical model can be represented schematically as in Fig. 1.1. Here are some qualities necessary or desirable for such a model:

1. Consistency with experimental data

2. Consistency with possible ideal physical mechanism

3. Be adapted to the needs of a user, in that it can predict useful parameters from input measurements accessible to the user

4. Have a minimum number of degrees of freedom (or independent input mea-

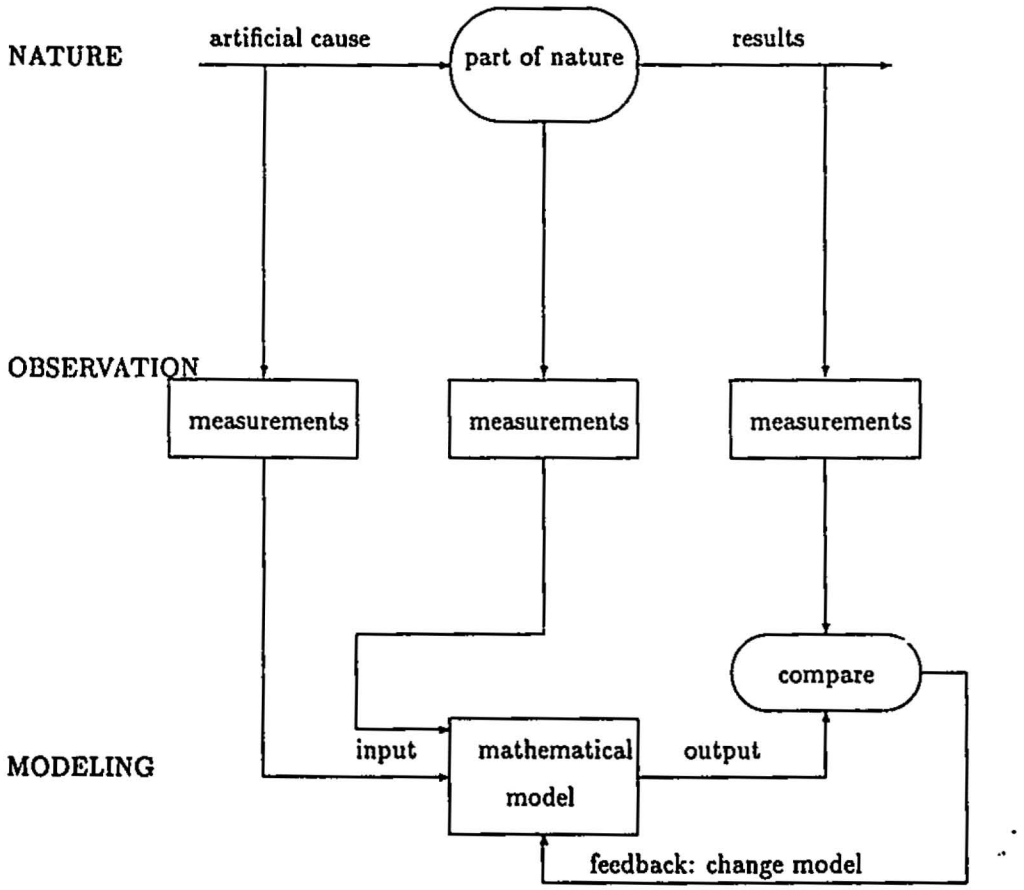


Figure 1.1: Relation between nature and mathematical models

surement)

With these criteria in mind, we immediately observe that the conventional sea clutter modeling in the literature mainly emphasizes conditions 1 and 3. Condition 1 is about the fitting of data which is the major work done in sea clutter modeling. Condition 3 has to do with the power of the model or the usefulness of the model. For engineering applications, this condition is particularly important since the ultimate goal of modeling sea clutter is to perform some operations like detection.

K-distribution is unique in that it is the only sea clutter model which considers condition 2. In other words, the model starts with or tries to relate the model to some reasonable physical mechanisms describing the backscattering process. This point is important in mathematical modeling because it gives confidence that the model is also valid outside of the regions where it has been verified.

Comparatively there is not too much work related to condition 4. The only progress on this condition is the observation that sea clutter is not suitably modeled by a pure noise process. Nevertheless, this observation is extremely important since it initiates all researches on sea clutter modeling. In other words, it forms the basis of modeling of sea clutter. Noise has a totally unpredictable, random behavior, and hence requires a very high number of degrees of freedom to describe it. Finding out that radar clutter is not merely a noise process is indeed a real breakthrough since not only the complexity (or degrees of freedom) can be reduced, but the performance will also be improved. Because of the significance of condition 4, we would raise a question : Do we really have a minimum number of degrees of freedom in sea clutter modeling by using all these conventional models ?

To make this question clearer, let us first point out one important common feature of all these sea clutter models. That is, all these models are based on statistical methods. In other words, sea clutter is considered to be governed by some kind of random

process. This is not an evidence but just an assumption. This assumption has been used in the modeling of radar clutter for more than fifty years, and surprisingly no one doubts the validity of the randomness assumption. To the best of our knowledge, this assumption does not have any rigorous physical explanation nor mathematical argument to support it. It is simply a result of the complex appearance of sea clutter to our naked eyes. The problems raised by this assumption is the introduction of unnecessary degrees of freedom. For example, to describe the linear acceleration of a rigid body, we just need to know the force acting on the object. If we include unnecessary factors like temperture, quantum effect, radiation from the rigid body,... etc, the performance of the model will be degraded. Another example is the motion of the earth; the generally accepted model of this motion is a circular rotation around the sun, described by Newtonian mechanics. It is unlikely that anyone will model this motion by some random process since it again complicates the model for no purpose.

1.2 Dynamics and Statistics

Those who try to explain the world we live in always hope that in the realm of the complexity and irregularity observed in nature, simplicity would be found behind everything, and finally unpredictable events would become predictable. Note that simplicity and regularity are associated with predictability. For example, because the orbit of the earth is simple and regular, we can always predict when astronomical winter will come. On the other hand, complexity and irregularity are almost synonymous with unpredictability. That complexity and irregularity exist in nature is obvious. We only need to look around us to realize that practically everything is random in appearance, of course including sea clutter. Or is it? Clouds, like many structures in nature, come in an infinite number of shapes, every cloud is different, yet everybody will recognize a cloud. Clouds, though complex and irregular, must

on the whole possess a uniqueness that distinguishes them from other structures in nature. The question remains : Is their irregularity completely random or is there some order behind their irregularity?

Over the last decades physicists, mathematicians, biologists, and scientists from many other discipline have developed a new way of looking at complexity in nature. This way has been termed *chaos* theory.

Chaos theory is the theory of dynamical systems that generate paths of evolution that appear random to the naked eye and to many statistical tests. Although chaos puts a limit on long-term prediction, it implies predictability over a short term. Recent research tells us that chaos is quite common, and that in many cases random behavior is due to low dimensional chaos rather than complicated processes involving many irreducible degrees of freedom. A computer pseudo random number generator is the standard example. This is a deterministic dynamical system whose path of evolutions from a given seed is perfectly predictable, provided you know the dynamics and provided you know the initial seed. This indicates that all the researches on random signal processing using computer simulation are actually simulated by using chaos theory rather than a truly random process.

Is there anything useful that chaos theory has to offer radar clutter modeling, which is the question set for this thesis? In other words, we are interested in whether the clutter process can be modeled by chaos theory. To the best of our knowledge, this thesis is the first research work done on this topic. We are thus confronted with many fundamental and important questions. The first one that comes to mind is : Why do we have an interest in chaos theory? Since statistical models are familar by now, and we do not feel that this random modeling causes many serious problems, why do we need to pay attention to chaos theory?

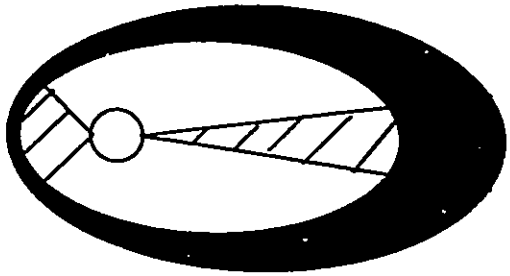
At this point we have to see how statistics comes into our business. We shall illustrate with a very simple hypothetical scattering model. Consider the example of

a billiard ball moving on a rectangular table with a convex obstacle. The ball moves in a straight line at unit speed and bounces off the obstacle and the edges with the angle of reflection equal to the angle of incidence.

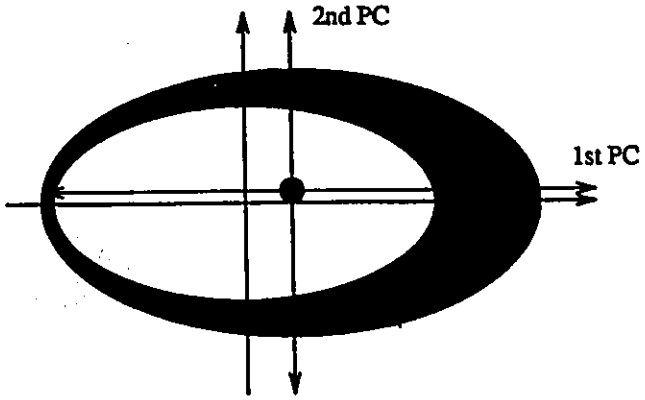
The configuration of the system is completely determined by the position and velocity of the ball. The phase space is the set of all configurations of the system. Each configuration in the phase space moves along a well-defined trajectory or orbit in the phase space. Phase space also has a probability structure : every reasonable (in probability language, measurable) set in phase space is assigned a probability - the probability of finding the system in that set. So the statistical methods tend to derive a probabilistic structure for the system. This is also the reason that statistics can only give us a partial description of a system being investigated.

To make this point clearer, let us look at the example of motion of the earth again. Now let us adopt the point of view of some observer who knows nothing about the underlying dynamics, and suppose that a person would try to describe statistically what is going on. Especially in models with more than three components, one tries to determine the directions of maximum variability. Those directions are called the principal components, which is now a widely used technique in signal processing. More technically, principal components are just the eigenvectors of the covariance matrix, i.e., the matrix of second-order moments.

But physically speaking, a statistician would not know where his time series came from and just computes its principal components to get something that pointed from one lobe to the other; principal components are defined up to sign : they indicate directions. In Fig. 1.2, we see the periodic motion of the earth around the sun. The speed of the earth at each point on its orbit is given by Kepler's Law of Areas, according to which equal areas are swept out by the radius vector in equal periods of time. In other words, the earth slows down considerably as it comes to aphelion and it speeds up as it nears perihelion. By symmetry, the mean (the first order statistic)



(a)



(b)

Figure 1.2: Periodic motion of planets

has to lie on the major axis of the elliptic orbit and it will lie closer to the part where the concentration of points lying equal intervals of time apart is bigger. Thus if you do not know what law is governing the motion of the earth, i.e, the dynamics, you would just take successive observations and see that the cloud of observations is much denser around the aphelion. So the time mean is closer to this point; furthermore the direction of the first principal component just points from this cloud where it is thickest to where the fewest points are found near the perihelon. Similarly, in the case of Fig. 1.2b the direction of the first principal component simply joins the two concentration maxima in phase space. Comparing to the study of the dynamics of the system directly, the abilities of statistical methods are very limited.

In our signal processing and communications communities, statistics has been a major analytical tools. The main reason for this is simply due to the noisy appearance of a signal in the system. Now, due to the discovery of chaos, our view of complexity and randomness are changed; we see no reason to restrict ourselves to statistics without seriously considering the possibility of chaotic signal processing.

Aside from the fact that dynamics provides a better description of the system than statistics, recent development in signal processing, communications, and also our sea clutter modeling problem tells us that Gaussian statistics is insufficient to solve all the problems by itself alone. In other words, all the traditional signal processing techniques such as power spectrum and Fourier transform based on second order statistics cannot satisfy the needs raised by the recent observations of nonlinearity. Non-Gaussian processes are getting more and more popular and common, and it implies that a reliable nonlinear statistical method is desperately needed in these areas. Unfortunately, nonlinear statistics is much more complicated than the linear one. This is actually not surprising if we look back at the history of deterministic mathematics, nonlinearity is really an extremely challenging problem. A simple example is the differential equation; solution of linear differential equations can be

obtained easily, whereas solving nonlinear differential equation is still an open problem. Thus, even though there is a great deal of work done recently trying to attack these problems by using higher order moments and cumulants, the results cannot be regarded as very successful.

First, the determination of a suitable higher order is not easy. Second order, which was suggested by Gauss, has many nice features. But, beyond this particular order, it seems that order determination is data-related. Second, if we model the process as a random process, everything becomes a random variable. This property makes the analysis even more difficult. Because of these reasons, research on higher order statistics has always been restricted to third or fourth order. Therefore , even if we want to follow this path, we should not expect too much.

To summarize what we have discussed, we suggest that the application of chaos theory in signal processing research is worthy of pursuit. First, it may reduce the degrees of freedom. Second, it is superior to statistics. Third, nonlinear statistical theory is not mature enough for us to use; however, nonlinear dynamics combined with topology and differential geometry may provide us with a usable mathematical technique to approach nonlinear signal processing problems.

1.3 Scope of this Thesis

After convincing ourselves why chaos theory is needed in modeling sea clutter (actually in signal processing), we would further consider how chaos theory can be applied. The answer to this question is the main objective of this thesis. Now let us first go back to the criteria for mathematical modeling to see what we should do.

The first condition reminds us that our model must be checked by experimental data. The sea clutter data used in this thesis were collected at a field trip to the East Coast of Canada by using a highly sophisticated radar system developed in

the Communications Research Laboratory at McMaster University, called Intelligent PIXel (IPIX) radar.

A traditional physicist's approach to model sea clutter may try to write down the scattering equation, and solve them with appropriate boundary conditions. This approach may start from the simple wave equation and the solution is obtained by the substitution of an appropriate scattering function. Later we may extend this to a nonlinear wave equation and study its behavior. But this is an exceedingly difficult problem. The effects of many different factors on the clutter process are still unknown to us; modeling this kind of phenonmena by coupled nonlinear partial differential equations is probably overambitious. This is the reason that clutter modeling is usually based on an empirical approach since the model must be definitive enough for us to use in specific circumstances. Another, more tractable, approach is to try and build a model directly from the data, without attempting to construct a universal model and then fitting the data to the model to check the validity. The model is directly obtained from the experimental data and usually satisfies condition 1 reasonably well.

Condition 2 concerns the underlying physical mechanism of the process. Conventional sea clutter modelings use the idea of random walk. More precisely, the backscattering of electromagnetic waves is viewed as a random motion. Almost all sea clutter models are based on this assumption; the difference may be just the number of scattering centers. In other words, the physical mechanism involves some incoming particles and some scattering centers to reflect those particles. This is one possible picture for what is really happening but it definitely is not the only one and perhaps not even a good one for explaining the real situation. However, it still supports the validity of the conventional statistical sea clutter models.

The question facing us is : Does chaotic modeling of sea clutter allow any reasonable physical explaination? Consider a string or a wave propagates toward a wall.

The reflection would be the exactly same wave form for elastic collision and some other wave forms with distorted characters such as frequency and phase for inelastic collision. Next consider a particle projected into a convex cavity; we can imagine that the particle will bounce inside the convex cavity and the trajectory of the particle will look quite random. If another particle is sent to the convex object with a slight deviation in the incidence angle from the first particle, we will expect a totally different trajectory for this new particle. The first particle may stay inside the convex cavity for just a while, but the second particle may stay inside the convex cavity for quite a long time. One may ask what is the purpose of these two examples. First, we feel that the backscattering of electromagnetic wave may be modeled by some deterministic process, since from the simple model of backscattering of a string, we observe that the backscattering process depends on the incoming wave and the scattering material. Hence the process should not be just a random walk. The second example tries to illustrate a point that scattering may be chaotic rather than random. The slight deviation meets the requirement of sensitivity to initial conditions. Though the trajectories look random, they follow a simple reflection principle. We therefore conclude that modeling sea clutter as a chaotic process is consistent with the possibilty of some ideal underlying mechanism.

Condition 3 is not a problem to chaotic modeling since it is a deterministic model. Condition 4 is in fact the one of the most interesting points to support the postulation of a chaotic model. Conventional sea clutter model considers the process as random and hence an infinite number of degrees is inherently assumed (all the entries in a stochastic process are random variables).

We therefore would like to construct a chaotic model for sea clutter using real-life radar data. The first thing that we probably should do is to investigate whether or not sea clutter allows a chaotic description. After that, we may then build a deterministic model for sea clutter. In Chapter 2, we will explain the basic theory of

chaos and the modeling technique based on chaos. To investigate whether sea clutter prefers a chaotic description to a random model or not is analyzed using two popular chaotic tests, namely, fractal dimension and a local divergence behavior. This task is undertaken in Chapters 3 and 4. In Chapter 5 we consider the modeling of sea clutter using conventional neural networks and a new structure called rational function neural network. In Chapter 6, a new neural network based on memory learning is used to model sea clutter. This model is then used to detect some small targets in sea clutter. Discussion of results and conclusions are presented in Chapter 7.

Chapter 2

Model Reconstruction of Chaotic Dynamics

2.1 Topological Description of Dynamical System

In Chapter 1 the term "dynamical systems" was used. What is a dynamical system? In simple terms, a dynamical system is a system whose evolution from some initial state (which we know) can be described by a set of rules. These rules may be conveniently expressed as mathematical equations. The evolution of such a system is best described by the phase space or state space of the system. An example of a dynamical system, a pendulum, and its state space, is given below.

Consider a pendulum that is allowed to swing back and forth from some initial state. The initial state can be completely described by the velocity and the position of the pendulum. The position of the pendulum at any time can be given by the angle. Under such an arrangement, Newtonian physics provides the equations (or rules) that describe the system's evolution from the initial state. Let us assume that the pendulum starts at some point other than the perpendicular position (the

lowest point). The pendulum will move downwards due to the gravitational force and will keep swinging back and forth as an exercise of exchanging kinetic and potential energies. Because there is some friction, there will be energy lost, and the pendulum will come to rest at the lowest point.

Apparently, the time evolution of the pendulum can be completely described by two variables, namely, velocity and angle. These two variables define the coordinates of the phase space. If one plots the velocity as a function of the angle, the trajectory will start off at some point in the phase space and eventually get closer and closer spirally to a stationary point which represents the lowest position of the pendulum. This particular point in the phase space is called an attractor. It "attracts" all the trajectories in the phase space. Apparently, the behavior of the dynamical system in question can be completely understood. Long-term predictability is guaranteed. The pendulum will always come to rest at the lowest point of its swing. Point attractors therefore correspond to systems that reach a state of no motion.

The above example of the pendulum only shows one form of attractors, that is, a point attractor. The next simplest form of an attractor is the limit cycle. A limit cycle in the phase space indicates a periodic motion. This phenomenon is not new to electrical engineers. For instance, limit cycles can be observed in the design of an finite-precision infinite impulse response (IIR) digital filter.

Another form of an attractor is the torus. The torus looks like the surface of a doughnut. In this case, all the trajectories in the phase space are attracted to and remain on the surface. Systems that possess a torus as an attractor are quasi-periodic. In a quasi-periodic evolution a periodic motion is modulated by a second motion, itself periodic, but with another frequency. The combination of frequencies will produce a time series whose regularity is not clear. The power spectrum, however, should consist of sharp peaks at each of the basic frequencies with all its other prominent features being combinations of the basic frequencies. Geometrically, a quasi-periodic

trajectory fills the surface of a torus, in the appropriate state space. An important characteristic of such an attractor is that when the two frequencies have no common divisor, any two trajectories which represent the evolution of the system from different initial conditions, and which are close to each other when they approach the attracting surface, will remain close to each other forever. This characteristic can be translated as follows. The two points in the state space where the trajectories enter the attractor can be two measurements which differ by some amount. Since these trajectories remain close to each other, the states of the system at a later time are going to differ to the same extent they differ initially. Thus, if we know the evolution of such a system from an initial condition, we can predict accurately the evolution of the system from some other initial condition. Again, in this case long-term predictability is guaranteed.

The above mentioned forms of attractors are "well-behaved" attractors and usually correspond to systems whose evolution is entirely predictable. Often they are called non-chaotic attractors. In mathematical terms, the above mentioned attractors are smooth submanifolds of the available state space. A very important characteristic of these attractors is that the long-term evolution of the systems they describe is not sensitive to initial conditions.

When one observes some irregular motion like the spectrum of turbulent motion, one realizes that there is motion at all frequencies with no preferred frequencies. This broadband structure of the spectrum indicates that the motion is nonperiodic (or strictly speaking is periodic with an infinite period). This nonperiodic motion was first observed by Lorenz [26] that a simple dynamical system was able to generate such a complicated behavior. In such a case, the trajectory in the phase space would be nonperiodic (never repeats itself) and would never cross itself (since once a system returns to a state it was in some time in the past, it then has to follow the same path). Thus the trajectory should be of infinite length but confined to a finite

area in the state space. This can only be the case if the attractor is not a topological manifold but rather a fractal set.

The fractal nature of such an attractor does not only imply nonperiodic orbits. It also causes nearby trajectories to diverge. As with all attractors, trajectories that are initiated from different initial conditions soon reach the attracting submanifold, but two nearby trajectories do not stay close to each other. They soon diverge and follow totally different paths in the attractor.

Divergence means that the evolution of the system from two sightly different initial conditions will be completely different. In other words, the state of the system after some time can be anything despite the fact that the initial conditions are very close to each other. Apparently, the evolution of the system is very sensitive to initial conditions. In this case we say that the system has generated randomness. We can now see that there exist systems which, even though they can be described by simple deterministic rules, can generate randomness. Randomness generated in this way has been termed chaos. These systems are called chaotic dynamical systems and their attractors are often called strange or chaotic attractors.

From the discussion above, we find that dynamical systems are conveniently described in terms of geometrical language like point, circle, torus and fractal. Although dynamical systems are often specified explicitly in terms of a real space, they can be generalized to exist on arbitrary manifolds. In a real space, a dynamical system with continuous-time evolution is specified by a system of ordinary differential equations

$$\dot{\mathbf{x}} = F(\mathbf{x}) \tag{2.1}$$

where $\mathbf{x} \in R^n$ is a point in the phase space and $F : R^n \rightarrow R^n$. This can be rephrased more geometrically by saying that F is a vector field on R^n; F associates a tangent vector with each point in the phase space. A particular solution to the set of differential equations translates into a particular curve through the vector field which

has a tangent vector at each point equal to that specified by the vector field. Such a solution curve or integral curve is denoted by the flow $\varphi_t(\mathbf{x}_0)$ which provides the value of $\mathbf{x}$ at time t given an initial condition $\mathbf{x}_0$. In the discrete time case, the dynamical system is described by a map $\varphi : R^n \rightarrow R^n$ and the flow is given simply by φ^i, where i indexes the discrete time.

The preceding discussion is easily rephrased in terms of manifolds. Let a compact manifold M be the phase space of a system. A dynamical system on M is a map $\varphi : M \rightarrow M$ for discrete time or a vector field for continuous time. The vector field associates each point in the manifold with an element of the tangent space at that point; it is a cross-section of the tangent bundle. The time evolution of the dynamical system is given by the flow $\varphi_t(\mathbf{x}_0)$, and an observable on the dynamical system is a smooth function $y : M \rightarrow R$.

We will need to describe the asymptotic behavior of a system. An invariant set S of a flow φ on a manifold M [32] is a subset of M defined by

$$S = \{\mathbf{x} \in M \mid \varphi_t(\mathbf{x}) \in S \ \forall \mathbf{x} \ and \ \forall t\} \tag{2.2}$$

The positive limit set $L^+(\mathbf{x})$ of a point $\mathbf{x}$ is the set of limit points that the flow approaches with an initial condition of $\mathbf{x}$:

$$L^+(\mathbf{x}) = \{\mathbf{y} \in M \mid \exists t_i \rightarrow \infty \ with \ \varphi_{t_i}(\mathbf{x}) \rightarrow \mathbf{y}\} \tag{2.3}$$

Here the t_i are a sequence of values of t, and the notation means that as this sequence tends towards infinity, the flow comes arbitrarily close to $\mathbf{y}$. Negative limit sets are defined for flows going backwards in time, and frequently positive limit sets are simply referred to as limit sets.

An attractor adds the notion of a domain of attraction to a limit set. A closed invariant set S is an attracting set if it has a neighborhood U such that $\varphi_t(\mathbf{x}) \in U$ for all t and $\varphi_t(\mathbf{x}) \rightarrow S$ (the flow comes arbitrarily close to all the members of S

) as $t \rightarrow \infty$. The domain or basin of attraction is the set of all initial conditions satisfying the above definition. We will usually consider attractors or limit sets in the following discussion, because the alternatives of settling to equilibrium or diverging are of much less interest in modeling complex real-life physical phenomena.

Now we have some basic ideas and mathematical tools to carry on our investigation. We will not give any more rigorous explanation of the concept of attractor; the reader is referred to [29] for a detailed discussion. To summarize what we have discussed, we know that a dynamic system is always characterized by an attractor which generates interesting behaviors that may be stationary, periodic, quasiperiodic, or chaotic. Traditional techniques of signal analysis are restricted to investigation in the time and frequency domains. They are based on the statistical procedure of deriving the covariance matrix of a time signal as well as the corresponding Fourier transform, the power spectrum. These techniques are very helpful in identifying stationary, periodic, and quasiperiodic attractor. But the idea of chaotic attractor has not been explored in signal processing. So the question is : How can we apply chaos theory? Narrowing down the question to our sea clutter modeling problem : How can we model the sea clutter phenomenon by chaos theory? Is it possible to do that?

One approach to the problem is to build a model for sea clutter from some well accepted physical phenomena like scattering, diffusion,... etc. This approach is quite powerful and common in mathematical modeling. For instance, the population model in ecology is derived by this principle, namely, competition of species. If we use this strategy to derive a model for sea clutter, we probably should start from the scattering process. In that case, we may get a partial differential equation, and by some mode decomposition, we may obtain a set of differential equations from the partial differential equation. Then we can analyze the set of differential equations to see whether chaos comes up or not, and hopefully we can find out in what parameter regions the process will become chaotic and determine the bifurcation process.

This approach is not the one taken in this thesis. First, the scattering model of electromagnetic waves in the literature is too immature to be used. For instance, scatterings from different materials like water, iron, ... etc are modeled by the same physical scattering model, and so we have no way to distinguish them. Second, the exact underlying process occuring in the scattering process is still unknown. In fact, it is not even very clear which factors are involved. This is why the majority of work done on this area relies on the use of experimental data in the so called empirical approach, since those hypothetical models are not ready for engineering applications yet.

Therefore, to develop a useful engineering oriented sea clutter model, we will follow the conventional line of thinking. That is, we try to build a model for the sea clutter based on experimental data. This is an inverse problem, which is very familar to engineers.

2.2 Embedded Dynamical System

A dynamical system is usually characterized by several components. For example, the pendulum example given in the last section has two components, namely, the position (angle) and the velocity. One may then suspect that to obtain the information about the dynamics, we need to measure all the relevant quantities. More troublesome may be the definition of relevant factors which is always unknown to us. In our case, to model the sea clutter dynamics, we may need to measure the wind speed, wave motion, temperature, different properties of the transmitted and received electromagnetic wave, ... etc. However, what are the real relevant quantities is still an open question.

If this is really what we need to do, even though dynamics can give us a better description of the system, dynamics approach will be too complicated to apply. Besides, we need to build a highly sophisticated radar system to perform the dynamical

study. That is, a radar which can measure all those relevant factors. In fact, data obtained by a radar system often take the form of a time series; a series of values sampled at regular intervals. Hence, we would like to build a model that reads a time series of data of the physical process, and from that to generate a model describing the future behavior of that process; that is, the dynamics. In other words, we hope that even though we just measure one component of a physical process, we can still use it to reconstruct the dynamics of that process.

At a first glance, such a quest appears to be ambitious and perhaps overoptimistic. How can we expect that measuring one factor can replace the measurement of all other factors? For example, a physical process is characterized by temperature, wind speed, atmospheric pressure and humidity, we cannot hope that such a process can be reconstructed by measuring just one of the factors. From the viewpoint of information theory, we can almost surely conclude that information will be lost if one component's measurement replaces many components' measurement. Fortunately, a recent theorem by Takens [37] suggests that our hope is not just a dream. In fact, the *Takens Embedding Theorem* states in principle that model reconstruction using just one component should succeed to a certain extent, and the reconstruction is independent of which components is used. Although the reconstruction cannot produce a model exactly the same as the original one, the reconstruction has the same behavior as the original system up to diffeomorphism.

Having established the type of model we intend to discuss in the previous section, we are ready to state the *Takens Embedding Theorem* and indicate the nature of the proofs. The system to be considered is a manifold M in which the system state is defined, a map $\varphi : M \to M$ (discrete time) or a vector field F on M (continuous time) , and an observable $y : M \to R$. We will show that M can be embedded in R^n through the observable y, that the limit set of the flow is reproduced, and that the dimension is unchanged by the process of embedding (this point will be discussed in

the next chapter). Fig.2.1 depicts a simple illustration of the embedding of a limit cycle.

To see how a model time series arises, consider some initial state x together with its subsequent states : $\varphi(\mathrm{x}), \varphi^2(\mathrm{x}) = \varphi[\varphi(\mathrm{x})], \varphi^3(\mathrm{x}) = \varphi[\varphi^2(\mathrm{x})], \ldots$ Suppose that the observable assigns the real number $y(\sigma)$ to the state σ; then a model time series $(a_1, a_2, \ldots, a_k, \ldots)$ is given by the numbers $y(\mathrm{x}), y[\varphi(\mathrm{x})], y[\varphi^2(\mathrm{x})], \ldots, y[\varphi^k(\mathrm{x})], \ldots$ Examples of this type of model abound. One such example is the damped, periodically forced pendulum, whose equation of motion is given by

$$\frac{d^2\theta}{dt^2} + Asin(\theta) + B\frac{d\theta}{dt} + Csin(\omega t) = 0 \tag{2.4}$$

where A, B, C and ω are nonnegative constants, θ is the angular displacement of the pendulum, and t is time. The phase space consists of all pairs (θ, v) where the angular displacement θ is an angle in radians and the velocity $v = r\dot{\theta}$ is a real number (r is the radius length of the pendulum). A natural observable is the kinetic energy given by

$$y(\theta, v) = v^2/2 \tag{2.5}$$

The law of motion or dynamics, φ is given by $\varphi(\theta_0, v_0) = (\theta_1, v_1)$ where (θ_1, v_1) is the result of integrating the differential Eq.(2.4). The results from $t = 0$ to $t = 2\pi/\omega, 4\pi/\omega, 6\pi/\omega, \ldots$ constitute a time series $a_0 = y(\theta_0, v_0)$, $a_1 = y[\varphi(\theta_0, v_0)]$, $a_2 = y[\varphi^2(\theta_0, v_0)], \ldots$.

The problem we want to attack is : If, for some dynamical system with time evolution φ_t, we know the functions $t \to y(\varphi_t(\mathrm{x}))$, $\mathrm{x} \in M$, then how can we obtain information about the original dynamical system from this? The next two theorems deal with this problem.

Theorem 2.1 *Let M be a compact manifold of dimension m. For pairs (φ, y), $\varphi : M \to M$ a smooth diffeomorphism and $y : M \to R$ a smooth function, it is a generic*

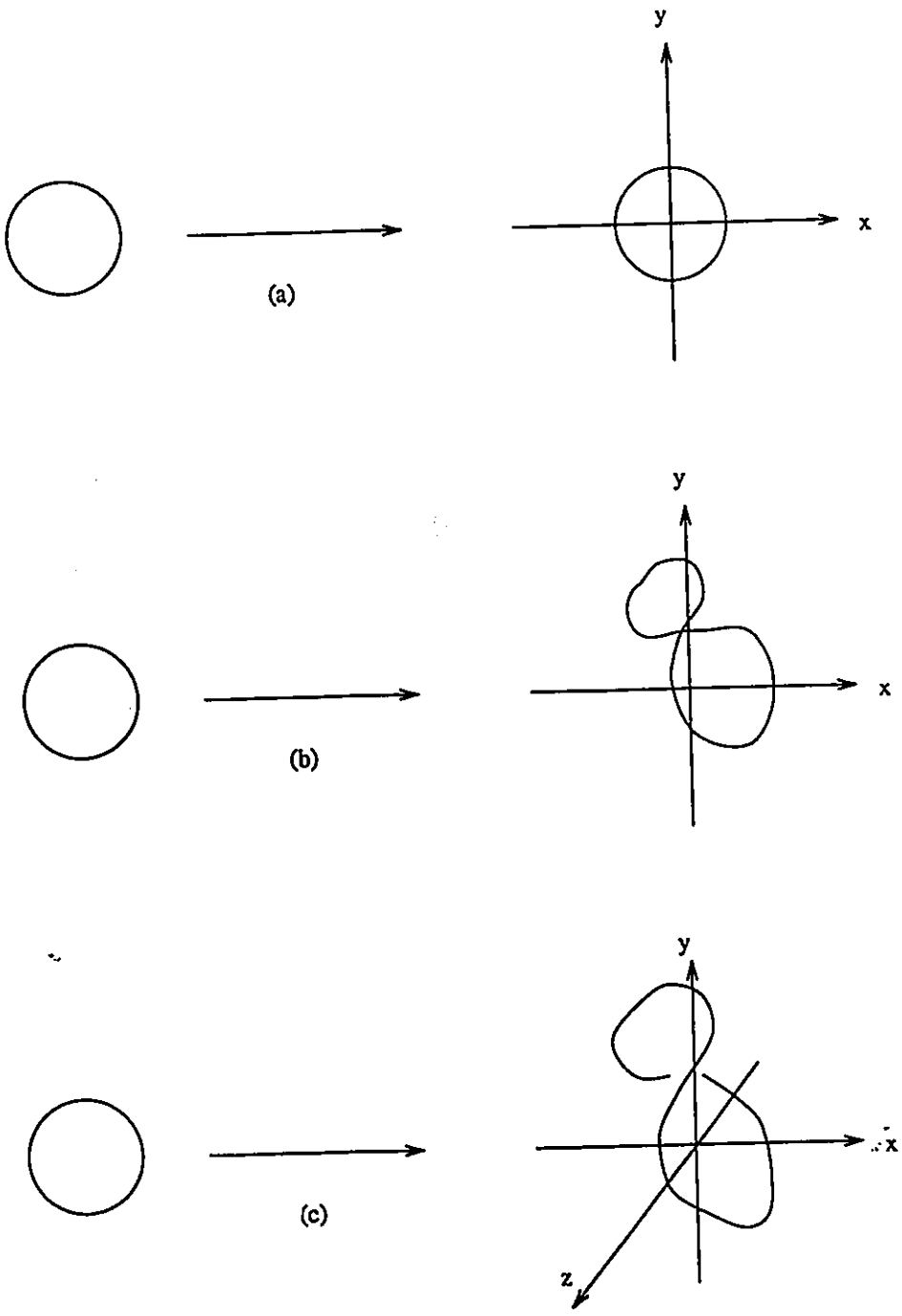


Figure 2.1: Embedding of a limit cycle: (a) embedding in R^2; (b) a mapping which is not an embedding; (c) the mapping in (b) becomes an embedding after a slight perturbation

property that the map $\Phi_{(\varphi,y)}(\mathrm{x}) : M \to R^{2m+1}$, *defined by*

$$\Phi_{(\varphi,y)}(\mathrm{x}) = \{y(\mathrm{x}), y(\varphi(\mathrm{x})), \ldots, y(\varphi^{2m}(\mathrm{x}))\} \tag{2.6}$$

is an embedding; by smooth we mean at least C^2.

To show that Φ is an embedding it is first necessary to show that it is an immersion, and this is the case if the covectors span the cotangent space, $T^*_F(M)$. Here Takens applies transversality to argue that for a generic M, y and φ, this will indeed be the case. This is the key concept of the theory. For only a pathological or trivial choice of M, y and φ will the derivatives of the iterates of the map be linearly dependent. If they are arbitrarily perturbed in the space of possible manifolds, functions, or observables, the linear dependence will be removed. The real world of thermal and quantum noise guarantees the experimentalist this perturbation.

After showing that the map is an immersion (a local property) it is necessary to show that it is injective (a global property). Takens makes this leap with a partition of unity, a technical tool that is frequently used for this purpose. The technical details are necessary to make the proof rigorous, but remember that the heart of the argument is the observation that the derivatives of the iterates of the map will generically be linearly independent. The theorem says nothing about just how independent the vectors are; we will later return to this important distinction between what is true mathematically (Φ is an embedding) and what is true experimentally.

The transversality arguments used are quite flexible; a wide variety of functions will serve just as well as time delays. All that is necessary is to find $2m + 1$ functions that are generically independent. In the next theorem, Takens shows that derivatives will work. Here φ_t again denotes the flow of F; this time smooth means at least C^{2m}.

Theorem 2.2 *Let* M *be a compact manifold of dimension* m. *For pairs* (F, y), F *a smooth vector field and* y *a smooth function on* M, *it is a generic property that the*

map $\Phi_{F,y} : M \to R^{2m+1}$, *defined by*

$$\Phi_{F,y} = \{y(\mathbf{x}), \frac{d}{dt}y(\varphi_t(\mathbf{x})), \ldots, \frac{d^{2m}}{dt^{2m}}y(\varphi_t(\mathbf{x}))\} \tag{2.7}$$

is an embedding.

From the last two theorems, it is clear how a dynamical system with time evolution φ_t and observable y is determined generically by the set of all functions $t \to y(\varphi_t(\mathbf{x}))$. In practice, the following situation may occur : We have a dynamical system with continuous time, but the values of the observable y is only determined for a discrete set of values of t. This is exactly the situation that occurs in our sea clutter modeling problem. Also, instead of all sequences of the form $\{y(\varphi_i(\mathbf{x}))\}_{i=0}^{\infty}$, $\mathbf{x} \in M$, we only know such a sequence for one, or a few values of $\mathbf{x}$. In this light we should know whether, under generic assumptions, the topology of, and dynamics in the positive limit set of $\mathbf{x}$ is determined by the sequence $\{y(\varphi_i(\mathbf{x}))\}_{i=0}^{\infty}$. This question is treated in the next theorem and its corollary.

Theorem 2.3 *Let M be a compact manifold, F a vector field on M with flow φ_t and p a point in M. Then there is a residual subset $C_{F,p}$ of positive real numbers such that for $\tau \in C_{F,p}$, the positive limit sets of p for the flow φ_t of F and for the diffeomorphism φ_τ are the same. In other words, for $\tau \in C_{F,p}$ we have that each point $q \in M$ which is the limit of a sequence $\varphi_{t_i}(p), t_i \in R, t_i \to +\infty$, is the limit of a sequence $\varphi_{n_i\tau}(p), n_i \in \mathcal{N}(integer), n_i \to \infty$.*

Note that this proof provides only a residual subset, not a more desirable full measure one. The set is almost certainly of full measure for any physical system, however proving this is more difficult and has not yet been done.

The preceding simple theorem, along with the earlier ones, implies a corollary that finally gives the central result on observing dynamical systems. We have seen

that discrete time maps on the manifold lead to an embedding in R^{2m+1} through the observable and time delays, and we have just seen that the conversion of a continuous time flow into a discrete time one by choosing a fixed sampling interval gives the same limit set. These results together imply that a discretely sampled data set embedded into R^{2m+1} has the same limit set as the real system.

Corollary 1 *Let M be a compact manifold of dimension m. We consider quadruples, consisting of a vector field F, a function y, a point p, and a positive real number τ. For such generic (F, y, p, τ), the positive limit set $L^+(p)$ is diffeomorphic with the set of limit points of the following sequence in R^{2m+1} :*

$$S_{F,y,p,\tau} = \{(y(\varphi_{k\tau}(p)), y(\varphi_{(k+1)\tau}(p)), \ldots, y(\varphi_{(k+2m)\tau}(p)))\}_{k=0}^{\infty} \tag{2.8}$$

Here diffeomorphic means that there is a smooth embedding of M into R^{2m+1} mapping $L^+(p)$ bijectively to the set of limit points of this sequence.

Takens Embedding Theorem therefore tells us that the trajectory of the embedded dynamical system is simply related to the trajectory of the initial system in its full phase space. In the sea clutter modeling problem, this means that if the experiment was able to simultaneously measure all the factors that are relevant to the sea surface backscattering process, a plot of the data would differ from the one generated by the trajectory of the embedded dynamical system by only a smooth change of coordinates. This change of coordinates will vary throughout the phase space, but in such a way that the two plots will differ only by local stretching : qualitative and quantitative measures of the behavior of the flows will be unchanged. After stating this important discovery, we can start our discussion on model reconstruction which is strongly based on the *Takens Embedding Theorem.*

2.3 Basis of Model Reconstruction

As we mentioned earlier, in model reconstruction we attempt to develop models for physical systems by seeking an algorithm that reads a time series of data observed from a physical process and, from that, generates a model to describe the future behavior of that process. This is called model reconstruction or reconstruction of dynamics. An advantage of the procedure to be described below is that, being highly empirical, it is uncontaminated by any particular physical preconceptions that may enter into the derivation of a theoretical model. Thus, comparison of the empirically reconstructed attractor with a theoretically predicted one offers the investigator a stringent test of the theory that makes the prediction.

At the most basic level, one applies model reconstruction to a time series $a_1, a_2, a_3, \ldots$ by finding an integer $N = 1, 2, \ldots$ such that N consecutive entries $a_{k+1}, a_{k+2}, a_{k+3}, \ldots, a_{k+N}$ determine the next entry in a fixed manner :

$$a_{k+N+1} = f(a_{k+1}, a_{k+2}, \ldots, a_{k+N}) \tag{2.9}$$

Of course, major problems with this procedure are immediately apparent. How do we know such a number N even exists? Even if we know that such a number exists, how do we find it? And granted that it is known, how do we find the function f?

Takens Embedding Theorem tells us that suppose that we have an m-dimensional manifold M as the phase space, then we may assign to each state $\mathbf{x}$ in M a point in Euclidean N-dimensional space given by

$$I(x) = \{y[\varphi(\mathbf{x})], y[\varphi^2(\mathbf{x})], \ldots, y[\varphi^N(\mathbf{x})]\} = (a_1, a_2, \ldots, a_N) \tag{2.10}$$

provided that N is greater than or equal to $2m + 1$. Since the assignment I defines a bijective correspondence between the phase space M and its image M', these two objects are entirely equivalent. In particular, the dynamic φ determines an equivalent

dynamic φ' on M' by the requirement that the equation

$$\varphi'[I(\mathrm{x})] = I[\varphi(\mathrm{x})] \tag{2.11}$$

be satisified. Upon using Eqs. (2.10) and (2.11), we see that for $(a_1, a_2, \ldots, a_N) = I(\mathrm{x})$, a point in M', the equation

$$\varphi'(a_1, a_2, \ldots, a_N) = (a_2, a_3, \ldots, a_N, a_{N+1}) \tag{2.12}$$

will hold with

$$a_{N+1} = y[\varphi^{N+1}(\mathrm{x})] \tag{2.13}$$

However, as long as it is in M', the point $(a_1, a_2, \ldots, a_N)$ uniquely determines the state x, which then via Eq.(2.13) determines a_{N+1}. Thus the point $(a_1, a_2, \ldots, a_N)$ of M' determines the coordinate a_{N+1} and we may write

$$a_{N+1} = f(a_1, a_2, \ldots, a_N) \tag{2.14}$$

for some uniquely defined function f. And clearly this function is the one required in Eq.(2.9). Finally, it follows from the *Takens Embedding Theorem* that this function is smooth.

Thus the *Takens Embedding Theorem* enables us to conclude that if our physical system admits a model with a m-dimensional manifold M as phase space and a generic transformation φ as the dynamic, then for the time series of any generic observable, an equation of the type in Eq.(2.14) will hold with N greater than or equal to $2m + 1$ and $f(a_1, a_2, \ldots, a_N)$ a suitable smooth function. In this way, the *Takens Embedding Theorem* carries us some way towards the resolution of the first of the problems outlined above, provided that we know that our physical system admits a model of known finite dimension.

We turn now to the interesting case in which we have reason to believe that our system does admit of a finite dimensional model although we do not know the dimension of the model. On the other hand, we assume that we do have available a very long

time series of data for an observable, as long a series as we may need, and we assume that it is a time series of the form given by $a_0 = y(\mathrm{x}), a_1 = y[\varphi(\mathrm{x})], a_2 = y[\varphi^2(\mathrm{x})], \ldots$ for some initial state x. We also assume that both φ and y are generic. In a favorable case, the initial state x may satisfy a condition of ergodic type that requires that the sequence of sucessive states $\mathrm{x}, \varphi(\mathrm{x}), \varphi^2(\mathrm{x}), \ldots$ visits every neighborhood of every state σ in the phase space M. In other words, the sequence $\mathrm{x}, \varphi(\mathrm{x}), \varphi^2(\mathrm{x}), \ldots$ fills up the phase space M. It then follows that the sequence of N-tuples $(a_0, a_1, \ldots, a_{N-1}), (a_1, a_2, \ldots, a_N), (a_2, a_3, \ldots, a_{N+1}), \ldots$ fills up the image phase space M'.

To illustrate this phenomenon, we consider a simple example. Let the phase space M be the unit circle in the complex plane,

$$M = \{z \mid z \textit{ complex and } |z| = 1\} \tag{2.15}$$

and we let the transformation φ be defined by

$$\varphi(z) = z \exp(2\pi i \alpha) \tag{2.16}$$

where i and π have their usual meaning and α is an irrational number, and for simplicity $0 < \alpha < \frac{1}{4}$. Then the sucessive states $z, \varphi(z), \varphi^2(z), \varphi^3(z), \ldots$ do indeed fill up the unit circle. For our observable we use the real-part function $y(z) = Re(z)$ so that we generate the model time series $a_0 = y(z), a_1 = y(\varphi(z)), a_2 = y(\varphi^2(z)), \ldots$. Now we may take $N = 2 \cdot 1 + 1 = 3$. As one may check easily, the 3-tuples $(a_0, a_1, a_2), (a_1, a_2, a_3), (a_2, a_3, a_4), \ldots$ fill up a planar ellipse in Euclidean 3-space and that planar ellipse is a distorted copy of the unit circle. In addition, once we have reconstructed this smooth copy of our original phase space, we may reconstruct the dynamic as the transformation

$$\varphi'(a_{k+1}, a_{k+2}, a_{k+3}) = (a_{k+2}, a_{k+3}, a_{k+4}), \tag{2.17}$$

where

$$a_{k+4} = f(a_{k+1}, a_{k+2}, a_{k+3}). \tag{2.18}$$

To find out the function f, consider the trajectory determined by

$$z_{n+1} = z_n \exp(2\pi i\alpha). \tag{2.19}$$

We obtain a time series of observations by using the real part function

$$s_n = Re(z_n) \tag{2.20}$$

Write $z_n = s_n + it_n$ so that s_n is the real part and t_n is the imaginary part. Then

$$s_{n+1} = s_n \cos(2\pi\alpha) - t_n \sin(2\pi\alpha) \tag{2.21}$$

and

$$s_{n+2} = s_n \cos(4\pi\alpha) - t_n \sin(4\pi\alpha) \tag{2.22}$$

Using $\sin(4\pi\alpha) = 2\sin(2\pi\alpha)\cos(2\pi\alpha)$, we may eliminate t_n from Eqs.(2.21) and (2.22) to obtain

$$2s_{n+1}\cos(2\pi\alpha) - s_{n+2} - s_n = 0 \tag{2.23}$$

Thus, the reconstructed space M' will lie in the plane defined by the equation

$$2a_2\cos(2\pi\alpha) - a_3 - a_1 = 0 \tag{2.24}$$

We have $\sin(2\pi\alpha) = [1 - \cos^2(2\pi\alpha)]^{1/2}$ and $s_n^2 + t_n^2 = 1$. From these two relations and Eq.(2.24) we see that M' also lies in the elliptic cylinder defined by the equation

$$a_1^2 - 2a_1a_2\cos(2\pi\alpha) + a_2^2 = \sin^2(2\pi\alpha) \tag{2.25}$$

In fact, M' is the planar ellipse equal to the intersection of the plane and the cylinder. Eq.(2.23) will hold also with n replaced by $n+1$, so that

$$s_{n+3} = 2s_{n+2}\cos(2\pi\alpha) - s_{n+1} \tag{2.26}$$

Solving Eq.(2.25) for $2\cos(2\pi\alpha)$ and substituting the result in Eq.(2.26), we obtain finally

$$s_{n+3} = (\frac{s_n + s_{n+2}}{s_{n+1}})s_{n+2} - s_{n+1} \tag{2.27}$$

which implies that the function $f(a_1, a_2, a_3)$ is given by Eq.(2.28); that is, by

$$f(a_1, a_2, a_3) = \frac{a_1 a_3 + a_3^2 - a_2^2}{a_2} \tag{2.28}$$

To summarize, we may say that the reconstructed phase space M' is the subspace of Euclidean N-dimensional space R^N filled up by the consecutive N-tuples of the given time series and that the reconstructed dynamic is given by

$$\varphi'(a_{k+1}, a_{k+2}, \ldots, a_{k+N}) = [a_{k+2}, a_{k+3}, \ldots, a_{k+N}, f(a_{k+1}, a_{k+2}, \ldots, a_{k+N})] \tag{2.29}$$

where the function f is the function which predicts the next entry in the time series. The *Takens Embedding Theorem* assures us that this function exists, and therefore in principle we may approximate on the reconstructed phase space by using the time series as data. In the favorable example above, for the sake of simplicity, we were able to find the function f directly from our knowledge of φ rather than from an approximation algorithm like the one to be described in the Chapters 5 and 6.

There are two conclusions to be drawn from the above example. First, we can indeed choose any component of the process to perform the dynamics' reconstruction; that is, any factor like the real part, imaginary part, amplitude, or phase component. This point is rather intriguing since our radar data are stored in complex form (which is a result of coherent processing), and this method tells us that we can just use either component. The interesting point is that using only one component for this modeling technique, we can perform the detection of a small target, a problem in which many people believe that both the amplitude and phase are required. This is a problem of extracting information from data; a probabilistic approach obviously is not as efficient as the dynamical method. This observation also opens a new line of

thinking: extracting more information does not necessarily rely on hardware but may also be possible through software. More precisely, radar researchers have a tendency to try to construct more advanced, sophisticated radar systems to achieve better performance. As in the above example, noncoherent radar is deemed not capable of detecting small target in sea clutter. So they look for a coherent radar which can provide more information. This approach is of course a feasible one; but the tradeoff is the higher expense of building a more complicated radar system. Dynamic theory tells us that a simple noncoherent radar using just the envelope may also have a good performance, provided we know how to extract the information from the data!

The second point is that an embedded dynamical system can indeed extract the underlying dynamics from the data. In the example, the dynamical attractor is a limit cycle and a diffeomorphic system is reconstructed from the data. The only problem that we may face (it does not appear in this example) is the construction of the map f. In this particular example, a direct construction is possible because it is simple and some prior information is given. However, this is not the case in general. In our sea clutter modeling problem, we have no existing knowledge on this modeling strategy and hence no prior assumption should be imposed to bias the model. The derivation of f is then the major and the most important part of this approach, since we need a model to perform further research like detection. We will solve this problem by the use of a neural network. We will show in Chapters 5 and 6 that it is a very powerful tool in extracting chaotic dynamics.

Before we can apply this novel model reconstruction technique to our sea clutter problem, one assumption of this techniqe must be justified for the sea clutter data. This assumption is the finite dimensionality of the model. That is, we have assumed that the system should admit a finite dimensional model and this dimension determines the number of variables involved in the process. Determining the dimension of the sea clutter attractor is covered in the next chapter.

Chapter 3

The Search for Attractors in Sea Clutter

3.1 Distingishing Random and Deterministic Systems

In order to illustrate the type of problem we want to attack in this chapter, we first discuss an example. Suppose we have some physical apparatus which, once it is started, gives a number as experimental output after each unit of time. So the output of running the experiment once is a bounded sequence $\{a_i\}$. For the moment we assume that in such sequences $A = \{a_i\}$ of experimental data, i runs through all natural numbers.

We are interested to know whether, on the basis of such a sequence, our apparatus is deterministic or not. Of course, to make this question more precise, we need a definition of a deterministic system. We say that a system or an apparatus as above, admits a smooth deterministic model if there is a smooth differential equation, or vector field, $\dot{\mathbf{x}} = F(\mathbf{x})$ on the phase space R^n, that is, $\mathbf{x}, \dot{\mathbf{x}} \in R^n$ and $F : R^n \to R^n$

smooth, and a smooth function $Y : R^n \to R$ such that

1. for each observed sequence $A = \{a_i\}_{i=0}^{\infty}$ of experimental data there is a point $\mathrm{x}_0 \in R^n$ such that $a_i = Y[\mathrm{x}_0(i)]$, where $t \mapsto \mathrm{x}_0(t)$ is the solution of the above differential equation starting at x_0, that is, with $\mathrm{x}_0(0) = \mathrm{x}_0$;

2. for each $\mathrm{x}_0 \in R^n$, the solution $\mathrm{x}_0(t)$ for $t \geq 0$ is bounded.

In this case we call the system smoothly deterministic. If only for some given sequence $A = \{a_i\}$, (F, Y, x_0) as above can be constructed such that $a_i = Y[\mathrm{x}_0(i)]$, we say that Y admits a smoothly deterministic explanation. The first condition is very obvious. It is the definition that we used in Chapter 2. The second condition may look strange at a first glance, but it is indeed necessary. Let $B = \{b_i\}_{i=0}^{\infty}$ be some bounded sequence. Then there is a smooth function $f : R \to R$ such that for all $i \in N$, $f(i) = b_i$. Now, taking as differential equation $\dot{x} = 1$, $x \in R$ and as initial point $x_0 = 0$, we see that, without the condition of bounded positive integral curves, any bounded sequence would admit a smoothly deterministic explanation.

Suppose next that the system is not deterministic, or rather, to be more explicit, that in the sequence $\{y_i\}_{i=0}^{\infty}$ of experimental data, each y_i is a random variable, independent of the other $y_i's$. We would expect that the data sequence cannot admit the above definition. In other words, there is a subset belonging to the space, which contains the set of all possible sequences of experimental data, is of measure one that no element of that subset admits a smooth explanation.

The conventional way to attack this problem is by transforming the experimental data into a power spectrum. It is the most popular technique in signal processing and it is good for the analysis of periodicity and quasiperiodicity. However, for other aspects of the data it is less satisfactory; for example, there is no way to tell from the power spectrum the difference between a signal generated by a strange attractor and a random signal. Therefore, it is very important to find some invariant

which can distinguish between random and deterministic processes. Starting at the early eighties, physicists and mathematicians began to use an invariant called fractal dimension to achieve this goal.

We shall explain what a fractal dimension is in a later section of this chapter. For now we outline the procedure for distiguishing between a stochastic system and deterministic system :

1. Measurement of a system The starting point is the measurement of a signal from the experimental system, $a(t)$. In our case, the system consists of a radar and a sea surface for reflection. In order to prevent artifacts from the digitization process, the signal must be sampled at a rate equal to twice the highest frequency present.

2. Phase Space Reconstruction Since the only information we have about the system is an experimental data sequence obtained through a physical apparatus, the first thing we need to do is to construct a phase space of the underlying dynamics. There is no fixed way to perform the reconstruction. Basically, we can use either differential or difference equations for model reconstruction. The only requirement is that the chosen variables should represent a phase space of a dynamical system. In other words, for an N-dimensional system, we should have N artificial variables to represent the original system so that the constructed system can be diffeomorphic to the old one.

3. Dimension Analysis Dimension of a dynamical system strictly relates to the number of independent variables of the system. In later sections of this chapter we will consider the rigorous definition and the measurement of dimension of an object given a set of points from the object. There are two reasons for the determination of dimension. First, the *Takens Embedding Theorem* holds under the assumption that the dimension of the underlying manifold is finite. Thus,

a finite dimension is crucial in showing the possibilty of using the dynamical approach for experimental data. Second, dimension can determine the number of dynamical variables involved in the expermental data received. Although the exact dimension of the attractor is not strictly equal to the number of dynamical variables, the number of dynamical variables needed can be guessed by knowing the dimension of the attractor.

4. Dynamics Estimation We can regard model reconstruction as a procedure for constructing algorithms by neglecting initial terms in the time series, by extending the series far enough, and by choosing a suitable level of resolution. We determine which subspace M' of Euclidean N-space is being traced out by the resulting N-tuples. Then, on that space one fits approximations of the predictor function to the data provided by the time series. The resulting system will reproduce an attractor from a possibly unknown finite-dimensional model for the physical process generating the time series.

3.2 Measurement of Sea Clutter

The Communications Research Laboratory (CRL) of McMaster University is conducting a programme of research to develop improved marine radar techniques for the detection and classification of icebergs and sea ice. In response to the equipment requirements of this programme, CRL has developed a research oriented radar system called Intelligent-PIXel (IPIX) radar, tailored specifically to the demand of research use.

Current marine radars utilize only the amplitude of the radar signal, and therefore ignore the information contained in polarization and phase. IPIX is an instrumentation quality radar system which has the following features. First, it is dual-polarized

and coherent. So it can provide information on more than just one component. This property is also useful to our dynamic research study since it provides a basis for comparison to our one-component dynamics reconstruction method. Second, IPIX has a built-in calibration equipment, which permits quantitative measurement to be made with high confidence, and a digital control system which allows experimental parameters to be easily varied, and once selected they remain accurate and stable. These equipments make the radar measurements highly reliable.

The IPIX radar system was installed at Cape Bonavista, Newfoundland, Canada in June 1989. Floating ice targets of all sizes commonly pass within radar range of this site during the Spring season and a wide variety of sea states can be observed throughout the year. The radar antenna is mounted at a height of about 25 metres above water, similar to that of a typical ship-mounted antenna.

During the field trip, various radar data such iceberg, fishing vessel and sea clutter was collected. Calibrations were performed which allow the processing software to relate any digitized signal value to an absolute power level at the receiver input. Our first job is to test whether the sea clutter is chaotic or not, and later in this thesis, we will apply the theory to detect some small targets such as a waverider.

To make a more convincing argument, we shall use five different types of sea clutter data collected at that field trip. These five data sets are chosen by the engineer who is in charge of the IPIX radar, and the author has no control on them. They are chosen according their variabilities to represent all the data sets collected in the field trips. Their properties are described as follows:

1. **D1** :

time	14 : 56 (06/07/89)
pulse repetition frequency	2000 Hz
polarization	HH
pulsewidth	200 ns
seastate	1.57 m

2. **D2 :**

time	12 : 48 (06/07/89)
pulse repetition frequency	2000 Hz
polarization	HV
pulsewidth	200 ns
seastate	1.30 m

3. **D3 :**

time	12 : 59 (06/07/89)
pulse repetition frequency	2000 Hz
polarization	VH
pulsewidth	200 ns
seastate	1.80 m

4. **D4 :**

time	19 : 51 (06/07/89)
pulse repetition frequency	200 Hz
polarization	HH
pulsewidth	200 ns
seastate	1.26 m

5. **D5 :**

time	18 : 25 (06/07/89)
pulse repetition frequency	200 Hz
polarization	HH
pulsewidth	200 ns
seastate	2.66 m

3.3 Phase Space Reconstruction

Before we can apply the nonlinear dynamical method to the data sequence, we have to reconstruct the state space of the system. The validity of the reconstruction is guaranteed by *Takens Embedding Theorem* discussed in the previous chapter. We will discuss the way to achieve this reconstruction in this section.

The purpose of the reconstruction is to reinterpret time signals as multidimensional geometrical objects. This allows us to use notions from geometry to describe and characterize a time evolution. This method is basically different from the classical approaches using autocorrelation functions and Fourier transforms, which are based on the underlying paradigm that all temporal oscillations may be decomposed into harmonic and periodic vibrations. The geometrical view of dynamical processes is based on the assumption that signals are generated by some deterministic and finite dimensional dynamical system, which is not necessarily the superposition of periodic oscillators. The observed signal can then be interpreted as the projection of a multidimensional phase space trajectory.

To illustrate the idea of phase space reconstruction, we will use a simple periodic oscillator for demonstration. The trajectory of a simple periodic oscillator in two-dimensional phase space is described by the equations $x(t) = a\sin(\omega t)$ and $\dot{x}(t) = a\omega\cos(\omega t)$. A simple trigonometric transformation shows that the velocity $\dot{x}(t)$ can be rewritten as : $\dot{x}(t) = a\omega\sin(\omega t + \tau) = \omega x(t + \tau)$. From this simple example we learn two things. First, for harmonic ocsillatory signals, we can replace one of the coordinates (or variables) with the other by looking at the other variable with a certain time delay τ. Second, there exist specific values of the time delay τ for which reconstructed variables are orthogonal.

Now, it should be clear that the heuristic idea behind the reconstruction is that to specify a N-dimensional dynamical system at any time, the measurement of any

N independent quantities should be sufficient. By the *Takens Embedding Theorem*, we know that any set of N independent quantities which uniquely and smoothly label the states of the attractor are diffeomorphically equivalent. One possible set of N such quantities is the value of the coordinate with its values at $N-1$ previous times, that is, the time delayed version of the time series $a_t, a_{t-\tau}, a_{t-2\tau}, \ldots, a_{t-N\tau}$ as we illustrate in the above simple example. Therefore, to reconstruct dynamics is simply a job of choosing a suitable time delay τ. In the ensuing flood of applications of this reconstruction idea, researchers found that some reconstructions are better than others. Usually a bad reconstruction is one that is not invertible, that is, points in the reconstructed phase space do not uniquely identify points in the original phase space.

The *Takens Embedding Theorem* tells us that the probability for making a bad choice is essentially zero; bad choices are those, for instance, that are commensurate with an aspect of the system's dynamics and hence restrict the region of phase space which is sampled. This is a mathematical statement however, not a practical one. In practical situations, especially in our radar or signal processing problem, noise which is unavoidable makes all reconstruction non-invertible and whether a reconstruction is good or bad is a question of degree. Also, the finite precision available to us will set a lower limit on the precision with which phase space can be viewed, and so trajectories that are mathematically distinct may be observationally identical. For τ near zero the trajectories will lie near the diagonal of the embedding space, and as τ is increased they will expand away from the diagonal. This implies that the best choice for τ is one that maximally separates nearby trajectories in phase space.

To obtain a good choice for τ, we first need to have a measure of goodness. Following the ideas of independence of state vectors, we can borrow some idea from statistics. To be more precise, we are looking for a delay so that the state vectors are as independent as possible. A good measure of this dependence in statistics would

be the widely used autocorrelation function between two random variables s and q :

$$I = \int s\, q\, p_{sq}(s,q)\, dsdq \tag{3.1}$$

If a time series of the observable $a(n)$ is given, we can simply apply the definition of autocorrelation function given above to the problem by choosing $a(n)$ as s and $a(n+\tau)$ as q. In this way, $a(n)$ and $a(n+\tau)$ can be choosen as independent as possible. A naive choice is then the value for which the autocorrelation function first passes through zero.

There are several reasons that we decide to use the autocorrelation function. First, the estimation of this function is well-analyzed, simple and unbiased. This means that the estimated value is reliable. Second, the dimension used in this thesis is the correlation dimension which is the second-order one in the information dimensions. Thus, the linear independence would be the most important and even the only information involved in the calculation of the dimension. Third, this method is less sensitive to noise. The autocorrelation function of the five data sets are plotted in Figs. 3.1-3.5. We can see that they have quite different behavior. D1 has its first zero-crossing at $\tau = 5$, and D4 is almost completely uncorrelated. The dynamic reconstruction will be performed for each data set according to its first zero-crossing value.

3.4 Attractor Dimension

Not only does the concept of attractor dimension serve to select the correct embedding dimension N, but it also serves as a means of illuminating the complexity of attractor structure. In this section we will be concerned with two definitions of dimension that are appropriate for attractors, the Hausdorff and correlation dimensions.

We begin with the Hausdorff dimension.To motivate the definition of Hausdorff dimension, we consider a cube in Euclidean 3-space. If we cover this cube with k

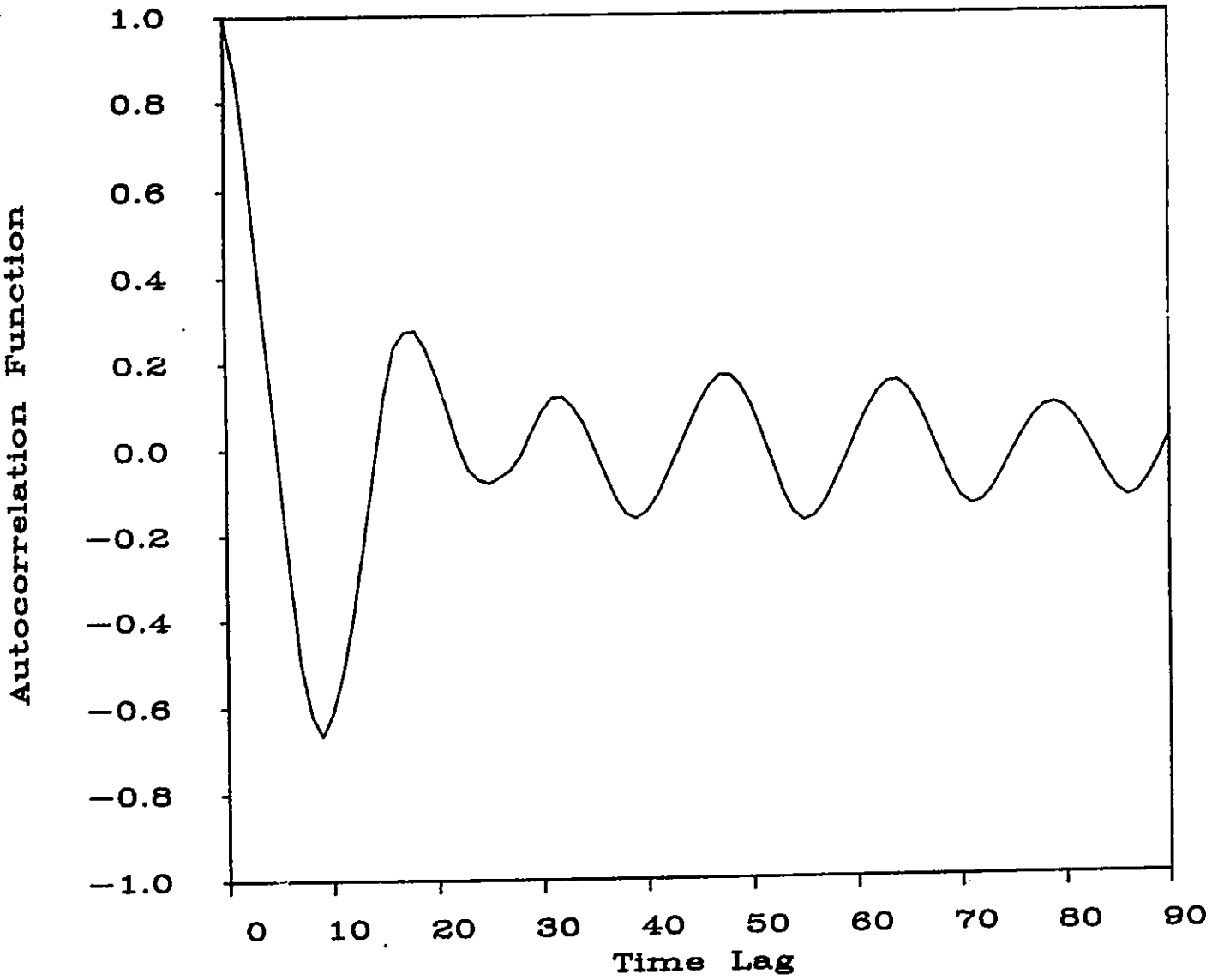


Figure 3.1: Autocorrelation function of D1

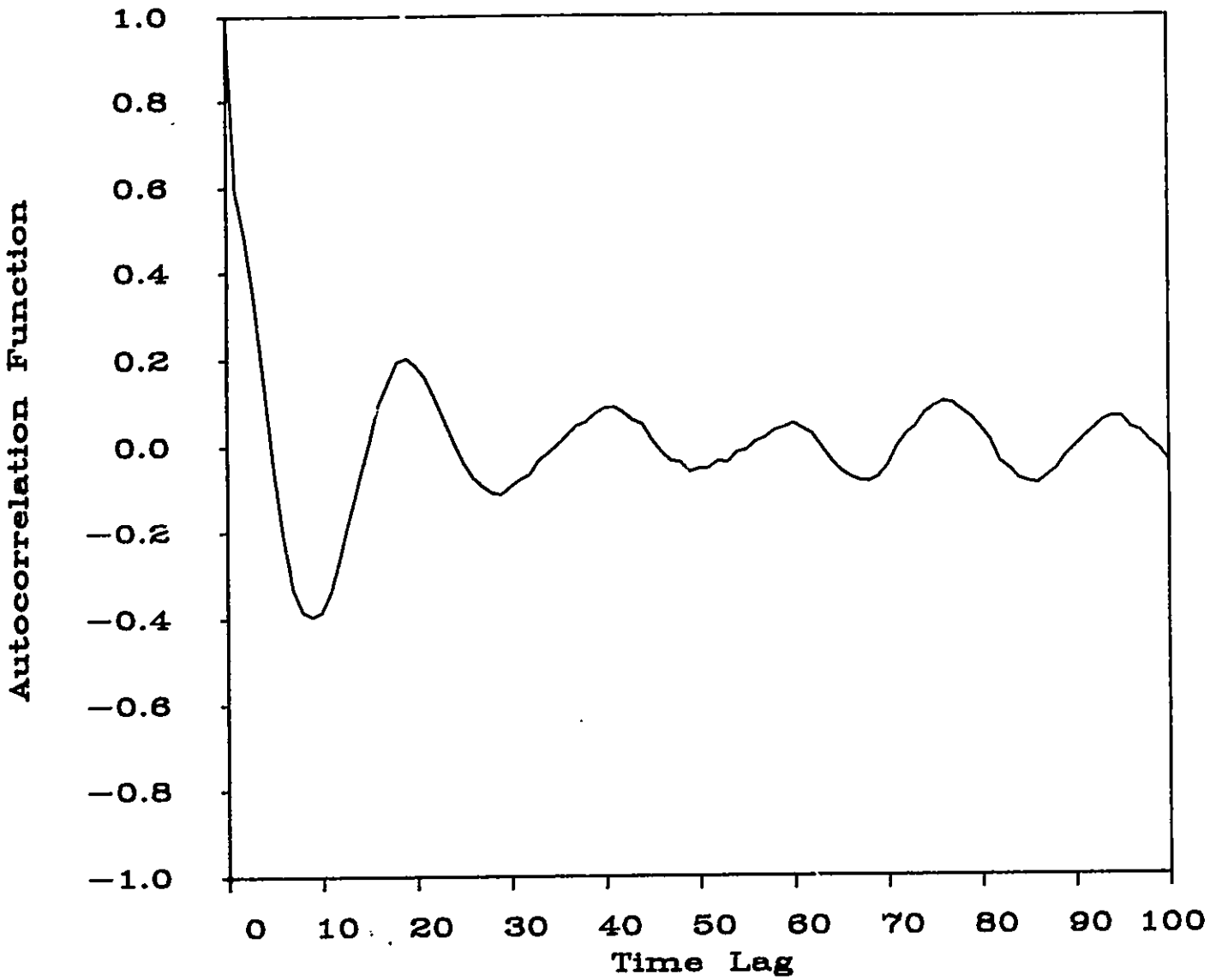


Figure 3.2: Autocorrelation function of D2

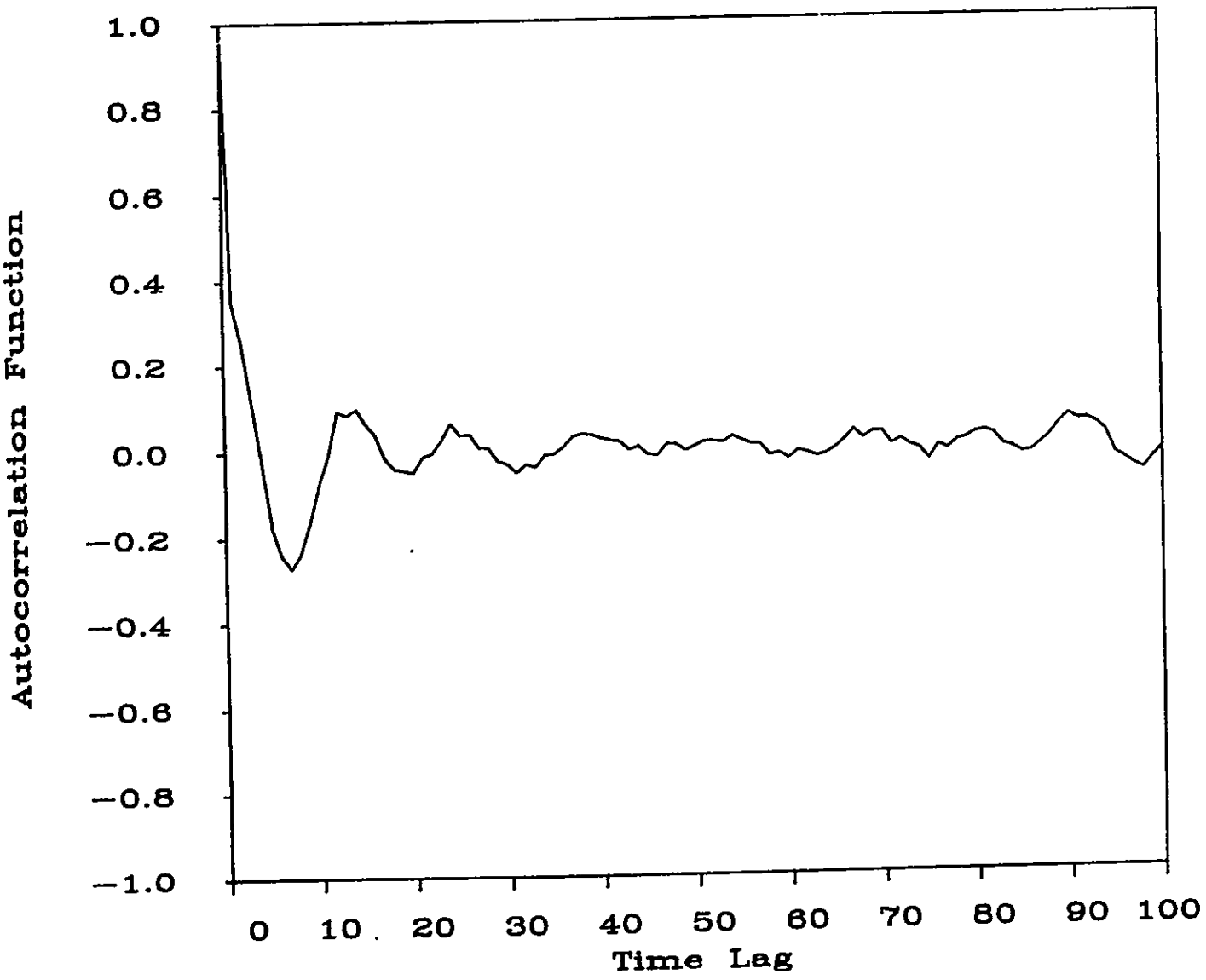


Figure 3.3: Autocorrelation function of D3

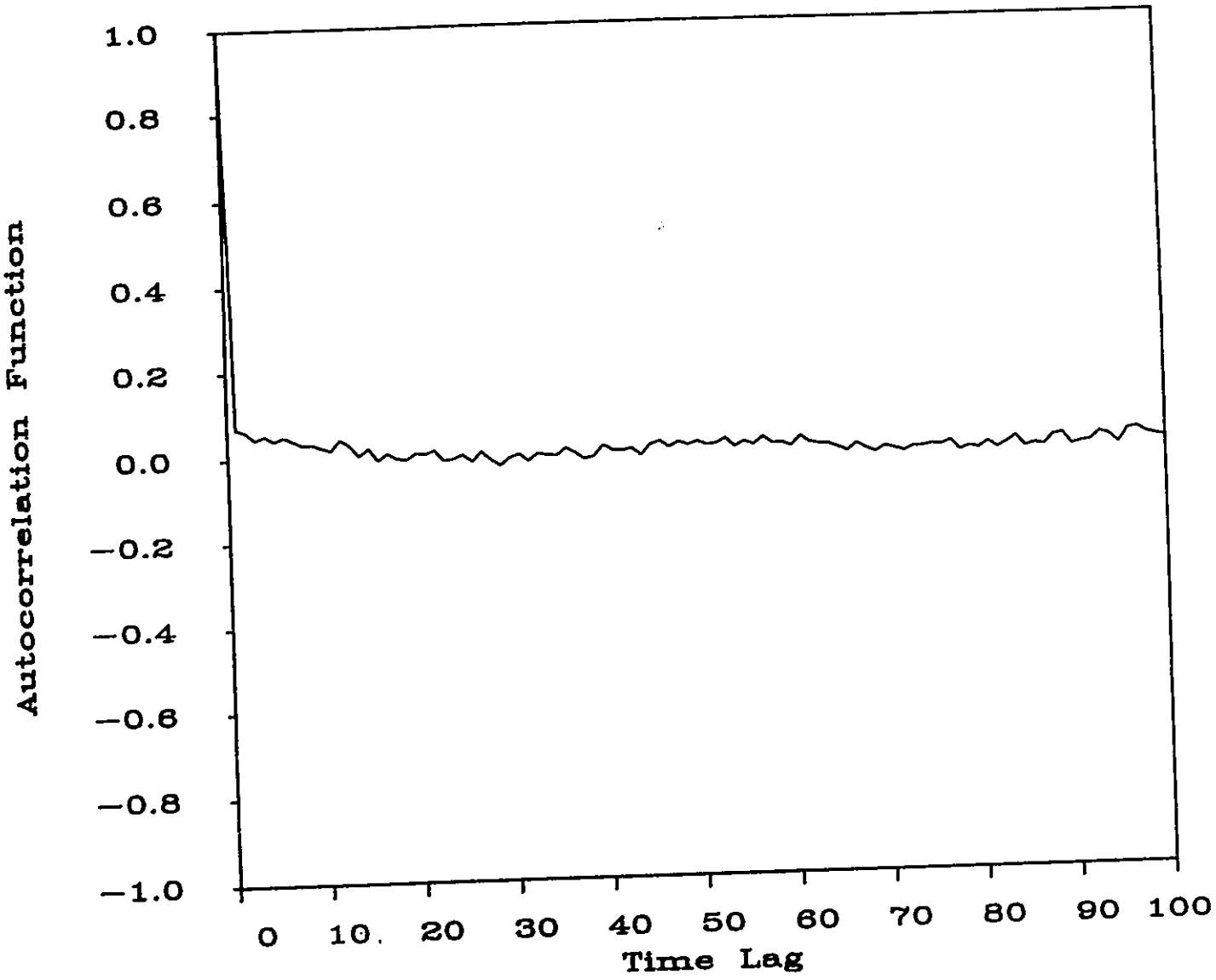


Figure 3.4: Autocorrelation function of D4

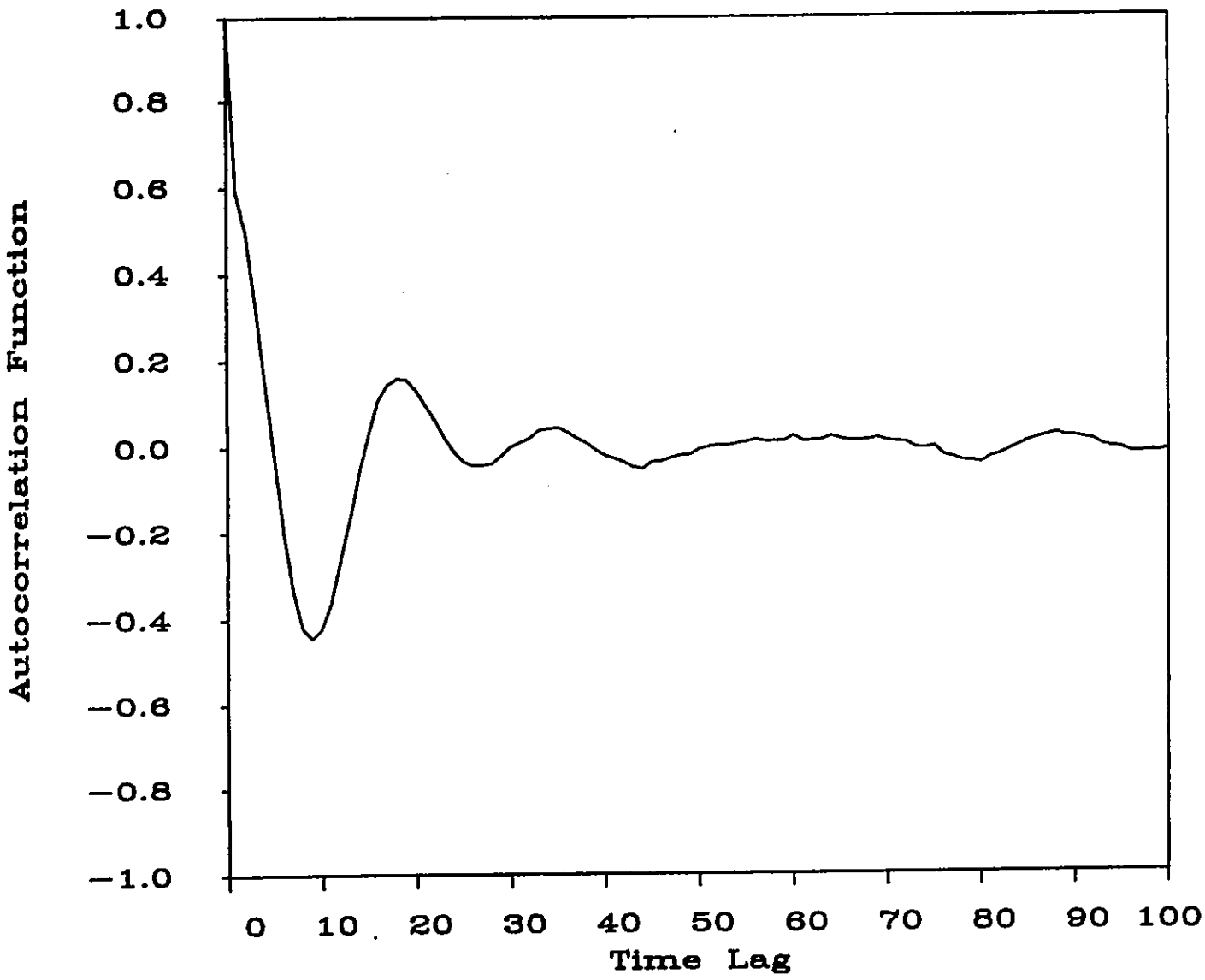


Figure 3.5: Autocorrelation function of D5

balls of radius r, we know that the number $k4\pi r^3/3$ is an overestimate for the volume of the cube; moreover, if k_r is the minimum number of balls of radius r necessary to cover the cube, then $k_r 4\pi r^3/3$ is a better overestimate of the volume of the cube. In fact, we know that the limit of these overestimates as the radius r approaches zero is the volume itself of the cube. Thus $\lim_{r\to 0} k_r r^3$ exists and is finite. It is easy to see then that the number 3 is a critical threshold for the cube. Mathematically, we can express this fact as

$$\lim_{r\to 0} k_r r^p = 0 \qquad for\ \ p > 3 \tag{3.2}$$

$$\lim_{r\to 0} k_r r^p = \infty \qquad for\ \ p < 3 \tag{3.3}$$

The same procedure is carried out for a square in the plane produces a threshold of 2. In general, for the hypercube in Euclidean n-space the threshold is n, and for a compact m-dimensional manifold, the threshold is m. Thus one is led to the following definition. Let J_r be the minimum number of balls of radius r necessary to cover the compact space A. If there exists a threshold d_f such that

$$\lim_{r\to 0} J_r r^p = 0 \qquad for\ \ p > d_f \tag{3.4}$$

$$\lim_{r\to 0} J_r r^p = \infty \qquad for\ \ p < d_f \tag{3.5}$$

then we say that d_f is the *Hausdorff dimension* of the space A. It is a routine matter to show that the number d_f determined by Eqs.(3.4) and (3.5) is given more directly by the equation

$$d_f = \lim_{r\to 0} \frac{\ln(J_r)}{\ln(\frac{1}{r})} \tag{3.6}$$

A serious difficulty still remains. The limits in Eqs.(3.4) and (3.5), may not exist and to resolve it we must introduce the limits inferior and superior. The limit inferior of a possibly nonconvergent sequence of real numbers is simply the smallest limit point of the sequence, and the limit superior of a sequence is the largest limit point. Because ∞ and $-\infty$ are allowable as such limits, the limit superior and the

limit inferior of a sequence always exist : they are equal if and only if the ordinary limit exists. We usually call the dimension with inferior and superior as lower (d_f^-) and upper dimension (d_f^+), respectively. Now, although these dimensions d_f^- and d_f^+ always exist, they are not always integers. For example, the Cantor set in Euclidean 1-dimensional space has the Hausdorff dimension $\frac{\ln(2)}{\ln(3)}$. The set is obtained by removing the open middle third from a unit interval, the open middle thirds from the resulting intervals, and so on ad infinitum. The set is dense, totally disconnected and closed. In some sense,it is fractured nature that the fractional part of the fractional dimension measures, and it is a similar fractured nature that is shared by most attractors.

To compute the Hausdorff dimension, assume that we are given a sequence of observables, $a(t)$. One can generate the complete state vector by using the reconstruction method described in the last section. For an N-dimensional phase space, a cloud or a set of points will be generated. The fractal dimension of this set can be estimated by covering the set by N- dimensional cubes of side l and determining the number of cubes $\mathcal{N}(l)$ needed to cover the set in the limit $l \to 0$ [13]. This is the so called box-counting algorithm and if this number scales as $\mathcal{N} \propto l^{-d}$ as $l \to 0$ then the scaling exponent d is an estimation of the fractal dimension for that set. In a $\log \mathcal{N}(l)$, $\log l$ plot the exponent d can be estimated by the slope of a straight line. If the original data sequence is random, then $d = N$ for any N (a random process embedded in a N-dimensional space always fills that space). If, however, the value of d becomes independent of N, (that is, it reaches a saturation value D_0), it means that the system represented by the time series has some structure and should therefore possess an attractor whose fractal dimension is equal to D_0.

The above procedure for estimating D_0 is a consequence of the fact that the actual number of variables present in the evolution of the system is not known and thus we do not know a priori what N should be. We must, therefore, vary N until we observe a structure which becomes invariant in higher embedding dimensions.

The above numerical approach to estimate the dimension of an attractor from a time series is, however, very limited. The reason is that an enormous number of points on the attractor is required to make sure that a given area in the phase space is indeed empty and not simply visited rarely. It has been documented [19] that a box-counting approach is not feasible for phase space dimension greater than two.

An alternative approach avoids the above metric thinking, and the idea is based on the use of probability. The most popular one in this approach is the *correlation dimension*, which is defined for a sequence of vectors $\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \ldots$ in the space A. The assumption is that this sequence is dense in the space A. Then one expects that the proportion of these points within a ball of radius r should be equal to the m-dimensional normalized volume of that ball, so that the normalized volume of the whole space A is equal to one. On the other hand, one also expects that the volume of a ball of radius r should be proportional to r^d as $r \to 0$. Accordingly, after fixing a point a_0 in A one is led to define first :

$$P_r = \lim_{n\to\infty} \left\{ \frac{number\ of\ integers\ k \leq n\ such\ that\ |\mathrm{a}_0 - \mathrm{a}_k| \leq r}{n} \right\} \tag{3.7}$$

so that P_r is the proportion of the sequence $\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \ldots$ that lies in the ball of radius r centered at the point a_0. Then since one expects that this proportion P_r is proportional to r^d,

$$P_r \sim r^d \tag{3.8}$$

we may define a dimension d by setting

$$d = \lim_{r\to 0} \frac{\ln(P_r)}{\ln(r)} \tag{3.9}$$

Next we modify the definition above to match that introduced in [17]. The definition we have just given really defines a dimension at the point a_0. The dimensions we seek are global; they must apply to the whole space. Therefore, we replace P_r with

its average over the whole space,

$$A_r = \lim_{n\to\infty} \left\{ \frac{number\ of\ pairs\ k,m\ \leq\ n\ such\ that\ |a_k - a_m| \leq\ r}{n^2} \right\} \qquad (3.10)$$

Then the definition of Grassberger and Procaccia for the correlation dimension is given by

$$d_c = \lim_{r\to 0} \frac{\ln(A_r)}{\ln r} \qquad (3.11)$$

Any difficulties with the existence of this limit may be eliminated by replacing the limits in Eqs.(3.10) and (3.11) with limits superior and limits inferior as appropriate.

To decide which of the modified limits is appropriate, we recall that Grassberger and Procaccia gave a heuristic argument to show that $d_c \leq d_f$. Here we replace that argument with a short but rigorous proof of the same fact and simultaneously determine which modified limits are appropriate in Eqs.(3.10) and (3.11).

We begin by expressing A_r in terms of the quantity given by

$$N(r,n) = \{number\ of\ pairs\ k,m\ \leq\ n\ such\ that\ |a_k - a_m|\ \leq\ r\} \qquad (3.12)$$

so that $A_r = \lim_{n\to\infty} \frac{N(r,n)}{n^2}$, provided that the limit exists. (Note that the counting in Eq.(3.12) includes the order of a pair, that is, double counting.) For brevity, write $p = J_{\frac{r}{2}}$. Then there exist p balls of radius $\frac{r}{2}$ covering the space A, and any two points in a single one of these balls are within distance r of each other. Some a_k may lie in the overlap of two or more balls; we choose one of these balls and say a_k definitely lies in that ball but not in the others. Let n_i be the number of integers $k \leq n$ such that a_k definitely lies in the ith ball. Then clearly we have

$$n_1^2 + n_2^2 + \ldots + n_p^2 \leq N(r,n) \qquad (3.13)$$

and

$$n_1 + n_2 + \ldots + n_p = n \qquad (3.14)$$

It is easy to see by using the methods of advanced calculus that the absolute minimum of $n_1^2 + n_2^2 + \ldots + n_p^2$, subject to the constraint of Eq.(3.14), is equal to $\frac{n^2}{p}$. Upon recalling that $p = J_{\frac{r}{2}}$ we see that

$$\frac{n^2}{J_{\frac{r}{2}}} \leq N(r,n) \tag{3.15}$$

or

$$\frac{1}{J_{\frac{r}{2}}} \leq \frac{N(r,m)}{n^2} \tag{3.16}$$

It follows that we have the inequality

$$\frac{1}{J_{\frac{r}{2}}} \leq \lim_{n\to\infty} \inf \frac{N(r,m)}{n^2} \tag{3.17}$$

or

$$\frac{1}{J_{\frac{r}{2}}} \leq A_r \tag{3.18}$$

where we define $A_r = \lim_{n\to\infty} \inf \frac{N(r,n)}{n^2}$ if the actual limit does not exist. By taking the natural logarithm of both sides and carrying out a few elementary operations, we see that

$$\frac{\ln(A_r)}{\ln(\frac{r}{2})} \leq -\frac{\ln(J_{\frac{r}{2}})}{\ln(\frac{r}{2})} \tag{3.19}$$

for which it follows immediately that Grassberger Procaccia inequality is valid in the form

$$d_c^{\pm} \leq d_f^{\pm} \tag{3.20}$$

if we define d_c^+ and d_c^- by setting

$$d_c^+ = \lim_{n\to\infty} \sup \frac{\ln(A_r)}{\ln(r)} \tag{3.21}$$

and

$$d_c^- = \lim_{n\to\infty} \inf \frac{\ln(A_r)}{\ln(r)} \tag{3.22}$$

The existence of the limit is a rather theoretical consideration. In general for typical attractors, the upper and lower dimensions will be identical. It is possible to prove

this rigorously for some sets. So henceforth we will not specify whether we mean the upper or lower limit in determining the dimension.

We should stress that the correlation dimension is not the only one available for experimental determination. There are other dimension measures such as the information dimension and the generalized dimension. There are several reasons that the correlation dimension is the most popular one. First, the ergodic characterization of dynamic system involves partitioning the phase space into cells and the probability of observing the orbit falling in the ith cell is given by $\frac{t_i}{T}$, where t_i is the number of times the orbit falls in the ith cell out of T points in the data sequence. The existence of such a probability is guaranteed by the existence of an ergodic natural measure, and in fact that probability is just the integral over the measure in the cell. It has been shown [18] that for finite T, there is a bias introduced in the calculation of dimension by using the naive estimate of probability described above, except for the correlation dimension. That is, the estimate $\frac{t_i}{T}$ can be used in computing the correlation dimension without producing yet another source of bias. Second, most dimension estimates are based on the assumption of the validity of the least squares relation as described in the computation of the Hausdorff dimension. This means that the deviations of data points from the least squares line are assumed to be normally distributed random fluctuations that arise because a finite sample of points are used to represent the attractor. To our knowledge this assumption is seldom acknowledged and, despite its wide use, has not been proved to be valid generally. However, this kind of statistical analysis has been performed on the correlation dimension [6], and has demonstrated the consistency and asymptotic normality of the correlation dimension.

Third, the determination of dimension using experimental data usually requires many data points. Correlation dimension is the only one that has been shown to work for a short data sequence. We are also aware of a negative result of using correlation dimension for small data sets [28]. Reference [28] has derived a formula

for the minimum number of vectors needed for correlation dimension calculation, in support of the assertion that large data sets are required, though no numerical results are provided in support of the formula. The central idea is that many loops of the attractor must be sampled to adequately represent its structure. This idea seems reasonable, but it strictly contradicts the use of a small data sequence. The problem in their derivation is the assumption that vectors are very closely spaced in time, which conflicts with the need for independent vectors and optimum sampling. The argument in [28] against small sets does not appear to overcome the other results [1] which support the use of dimension analysis on small data sets.

Having outlined the concepts of model reconstruction, attractors, and dimension, and having established the Grassberger Procaccia inequality, we are ready to test these concepts by applying them to the sea clutter data described in Section 3.2.

3.5 Correlation Dimension of Sea Clutter

The definition and elementary properties of the correlation dimension of Grassberger and Procaccia were given in the preceeding section. In this section we address its numerical determination from empirical data.

We begin with a set of points $\{\mathbf{a}_k \mid k = 1, \ldots, N\}$. It is in the nature of most strange attractors that pairs of trajectories that are initially close will eventually become temporally uncorrelated. However, both trajectories will still be on the attractor and remain spatially correlated. This cumulative correlation was defined by Grassberger and Procaccia as

$$C(r) = \frac{1}{N^2} \sum_{k,j=1}^{N} H(r - |\mathbf{a}_k - \mathbf{a}_j|) \qquad (k \neq j) \tag{3.23}$$

where H is the Heaviside function

$$H(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.24}$$

The norm$|a_k - a_j|$ may be any kind of norm measure. The Euclidean norm is probably used most often, as it is the usual way to calculate the distance between two points. While this norm has the advantage of being familiar, it is the most time consuming to calculate.

The relationship between the correlation dimension and the cumulative correlation function is based upon the power law Eq.(3.8) which is derived from the geometric considerations previously discussed. The result is valid for " small enough" values of r. There are several examples given in [17] that illustrate the region in which the power law holds. A determination of the correlation dimension is found by plotting $C(r)$ vs r on a $\log - \log$ graph. The region in which the power law is obeyed appears as a straight line, and the slope (which is an estimate of the correlation dimension) is found by fitting a least squares line to this part of the graph. If the sequence of estimates converges, then we have an estimate of the attractor dimension d_c.

The algorithm can be summarized as follows :

1. Form a matrix whose columns are the embedded dynamical vectors constructed from the experimental data by the method outlined in Section 3.3.

2. Form a matrix of the Euclidean distances between the columns. The matrix is symmetric, and hence only the upper triangular part is needed by the algorithm.

3. With this matrix available, the correlation matrix may be calculated directly from its definition.

4. The final step is to plot the results on *log-log* paper, and fit a least squares straight line to the lower part of the data. The slope of the line is then the

correlation dimension.

Assume that there are T data points in the data sequence, determining the correlation index for a set in N-dimensional space requires us to calculate Eq.(3.12). The Grassberger-Procaccia algorithm requires storing T vectors (TN numbers) and $\sim T^2N^2$ operations. Therefore, we use the optimized algorithm for correlation integrals proposed in [15] to compute the correlation. The algorithm simplifies calculations and reduces considerably their time requirements.

Note that the Grassberger-Procaccia algorithm is designed for multidimensional dynamical systems. It follows from *Takens Embedding Theorem* that we need not consider a set of N-dimensional vectors which are characterized by the collection of all the coordinates of the point in the phase space of the dynamical system, but just one coordinate. Another point is that in deriving the *log-log* plot, we need to increase the embedding dimension to let the correlation dimension converge. The computation of the distances between vectors for each embedding dimension will therefore produce many repetitions. The optimized algorithm in [15] is developed to take care of these two points and results in a much faster computation. Since the detailed development is quite long, we will not explain it here and refer it to [15]. There are also other algorithms such as the so called prism box assisted algorithm [38] which can reduce the time consuming problem for large T, but it does not optimize the computation procedures discussed above.

Now we are ready to apply the algorithm to our sea clutter data. Because sea clutter data come from a real experiment, it will be contaminated by noise. We will not do anything on this external noise for the following reasons. First, the dimension analysis itself has the ability to identify external noise. More precisely, those curves on the *log-log* plot will bend. Thus, the dimension will be higher in some regions and the effect is due to external noise. Second, when the signal to noise ratio is higher than

25 dB, it is believed that the correlation dimension estimation is valid [31]. Third, one may suggest to filter the sea clutter data to suppress noise. A common technique would be to use low pass filtering. We should remember that this concept is based on the paradigm of Fourier analysis, that is, the relevant information is contained in signals of a more or less well defined frequency. This approach is reasonable when all the relevant signals occur with frequencies below the cutoff frequency of the low pass filtering. When we have a broadband signal as in the case for chaos, we have to be very careful with the interpretation of the observed results. Intuitively, we would expect that the filtered time series will have a lower dimension than the unfiltered one since it has been cleaned from the noise which admits infinite degrees of freedom. In fact, the opposite is observed [3]. In this reference, an increase of the fractal dimension from 2.5 to 3.2 of the Duffing attractor has been observed. This is because if the underlying process for the time series is deterministic, low pass filtering is actually adding a new dynamical variable to the original motion. Thus, filtering of noise is not recommended in our sea clutter data analysis.

The results of the correlation dimension analysis for data set D1 - D5 are plotted in Figs. 3.6 - 3.15. In obtaining these results, 10,000 data points were used. The analysis performed on all five data sets shows an encouraging result, namely, saturation of the slope in the *log-log* plot. The fractal dimensions of the data sets analyzed are different. D1 has a dimension around 6.5; the dimension for D2, D3, D4, and D5 lies between 7.2-7.5, 7.9-8.2, 8.7-9 and 7.4-7.7. The dashed curves in Figs. 3.11 - 3.15 correspond to a Gaussian random process, which have been included for the sake of comparison with the real radar data. These curves were obtained by applying the correlation dimension analysis to a computer generated random Gaussian white noise sequence.

We may conclude from these five examples that the correlation dimension of sea clutter is between 6 and 9. However, it seems impossible to make any conclusive remark about the relationship between the correlation dimension and properties of

sea clutter data by simply observing this correlation dimension.

Care must be exercised in applying this technique; especially for real data. First, the number of data points used in the experiment is a very crucial factor for obtaining a reliable result. We believe that it is not necessary to have too many data points, but the number of data points used must be enough to allow the orbit to go through a neighborhood sufficiently many times as explained in [1]. Second, fitting the least squares line in the linear region must be done very carefully. This is the part where it is easy to introduce errors. The main problem is that the linear region may not be very clear. The shorter the linear region is, the easier it is to generate an error. Some methods have been suggested to take care of this difficulty, such as using the generalized dimension to determine a suitable linear region [32]. Fortunately, in our situation the *log-log* plot has a very smooth appearance, and the linear region does not seem to be very difficult to locate. In the process of fitting a straight line, we performed a least squares fit to various portions of the region that appear to be straight to our naked eyes. It was found that the fits were quite close. Another observation from these figures is that an immediate saturation does not happen. For instance, the estimated fractal dimension of D1 is about 6.5, but saturation does not occur until the embedding dimension is raised to 11 or 12. This may look a little bit strange at first glance. However, there is no contradiction to the theory implied here, since it does not violate the *Takens Embedding Theorem.*

In this chapter, we have introduced a very important concept, namely, the fractal dimension, to be used to study the behavior of a dynamical system. This invariant is so specfic that not only can it indicate the behavior of a dynamical system (periodic, quasiperiodic, or fractal), but it also tells us the number of dynamical variables involved in the process. It is believed that the dimension analysis is the most powerful analytic tool since the power spectrum. A special kind of fractal dimension, that is, correlation dimension, is applied to our sea clutter data. We have

made a very important discovery that sea clutter data behave differently from what we expect to observe from a purely stochastic process. Specifically, for the radar data analyzed in this thesis, sea clutter has a fractal dimension varying between six and ten, depending on the physical circumstances of the sea surface.

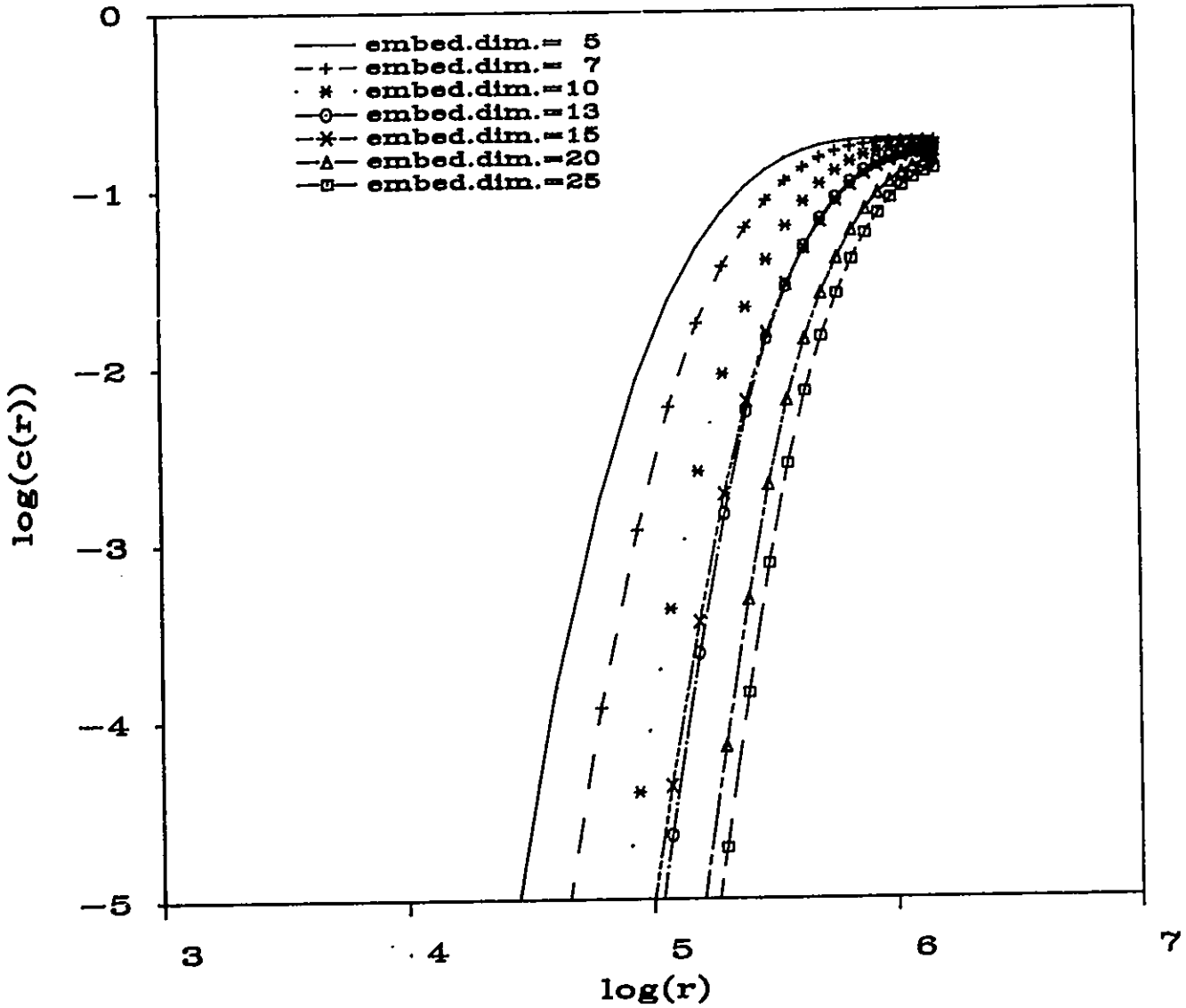


Figure 3.6: Log-log plot correlation integral of D1

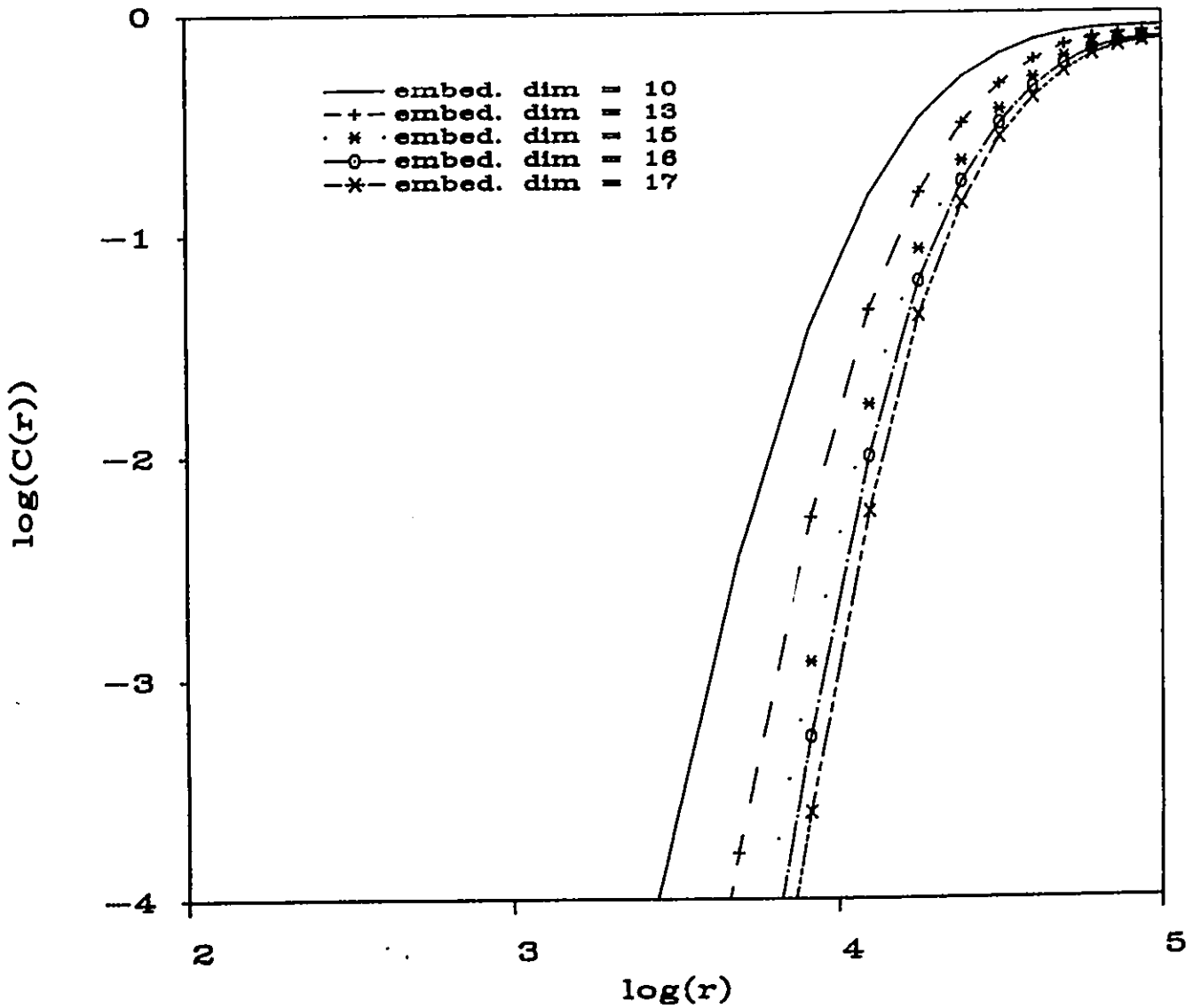


Figure 3.7: Log-log plot correlation integral of D2

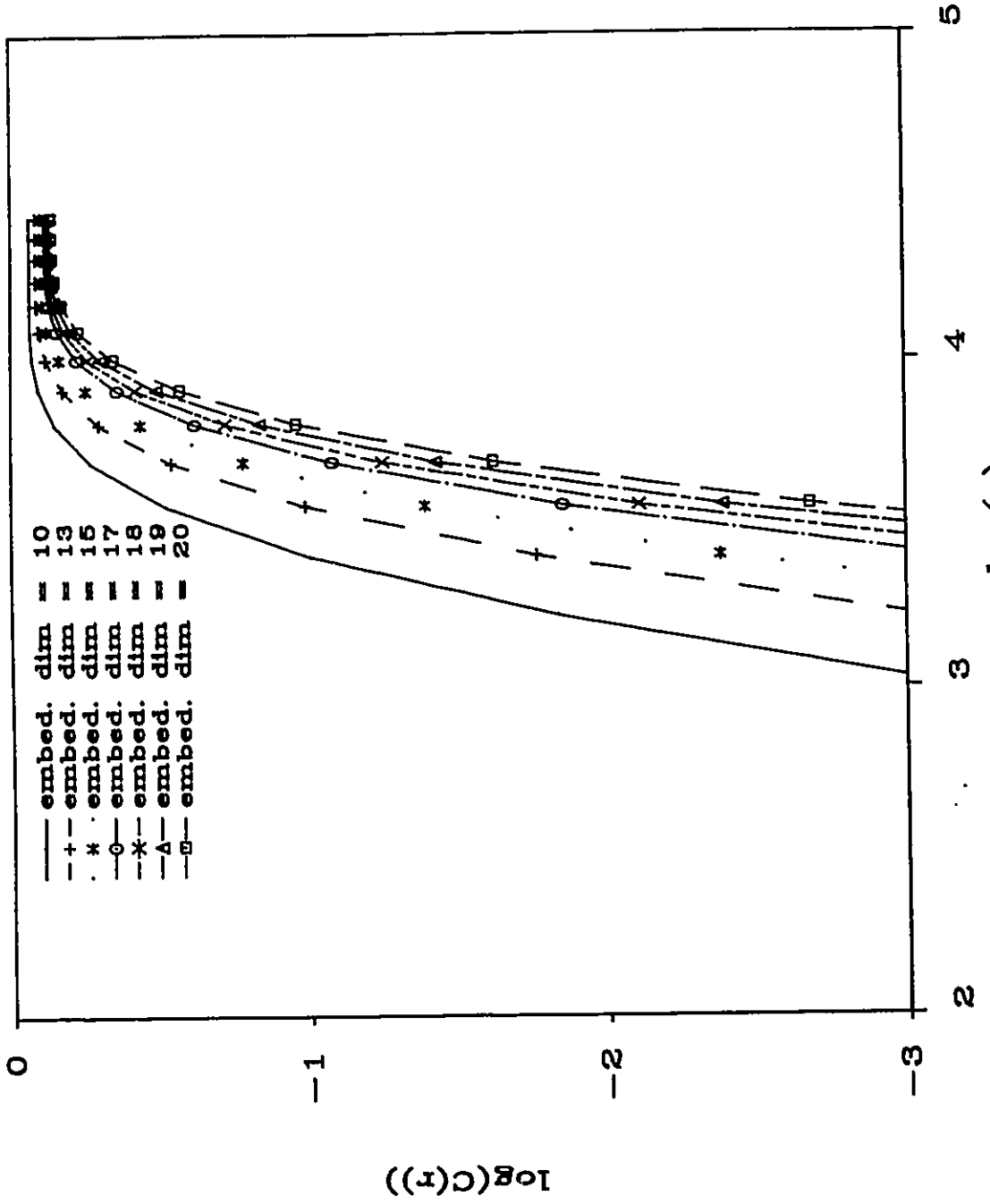


Figure 3.8: Log-log plot correlation integral of D3

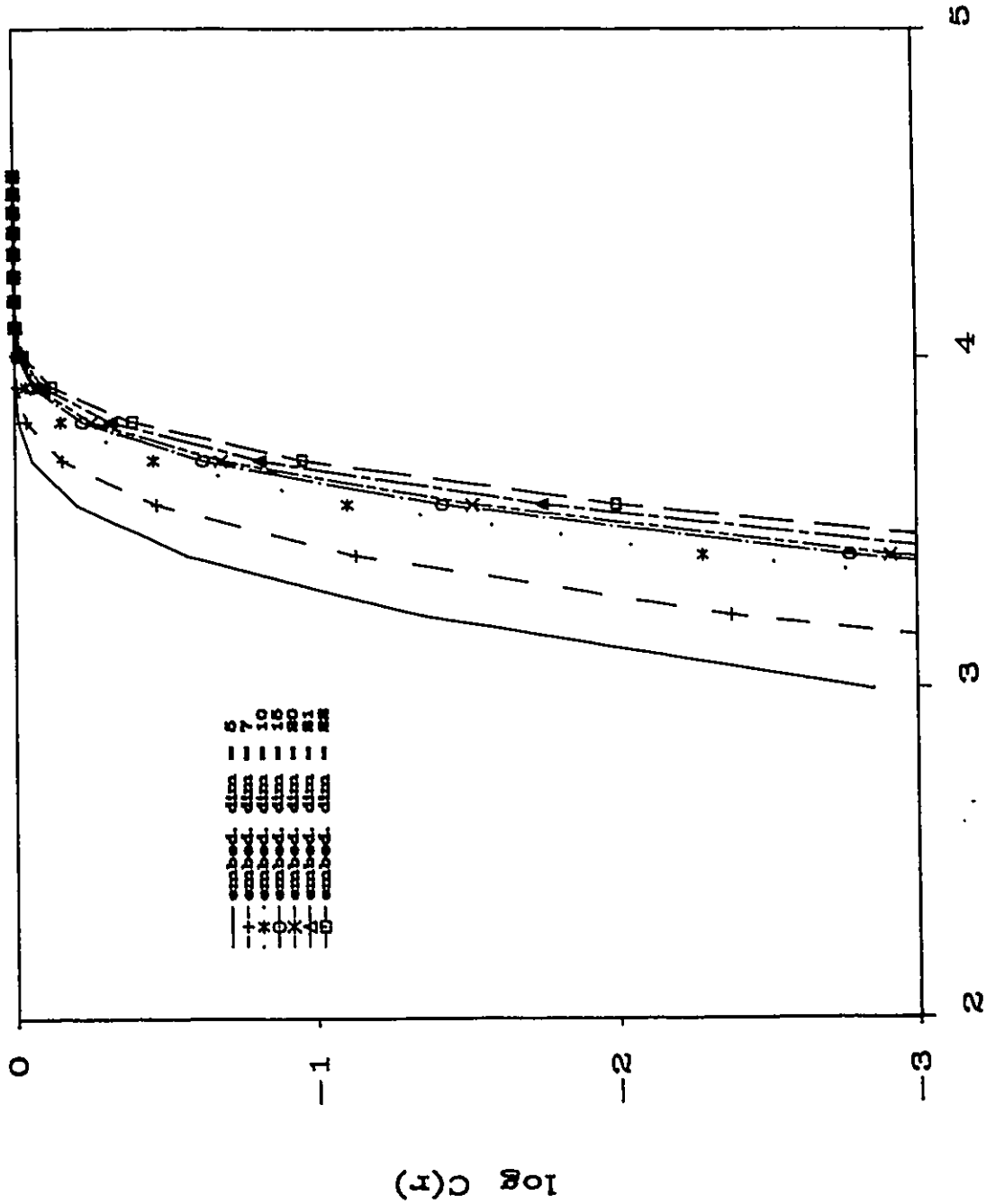


Figure 3.9: Log-log plot correlation integral of D4

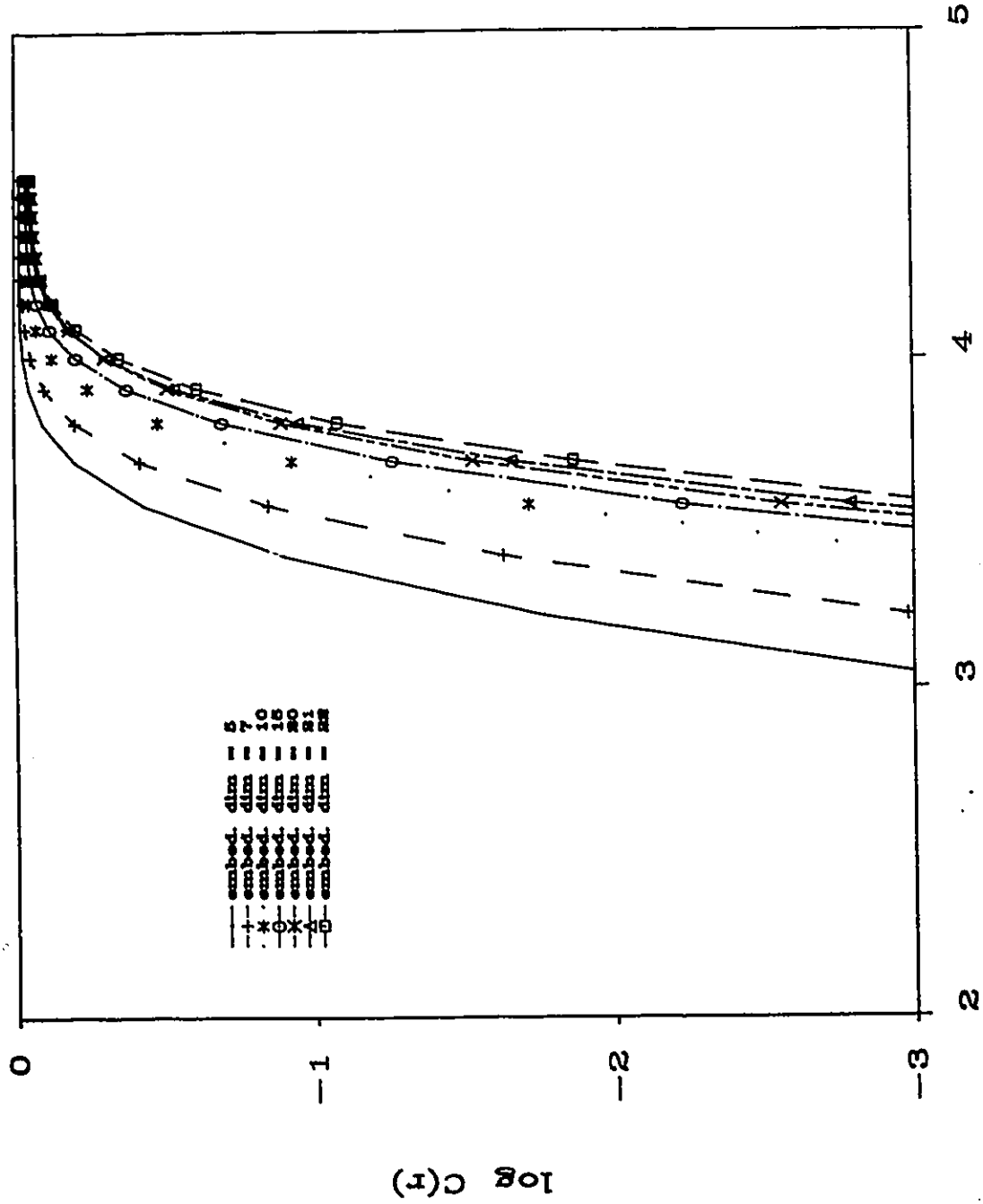


Figure 3.10: Log-log plot correlation integral of D5

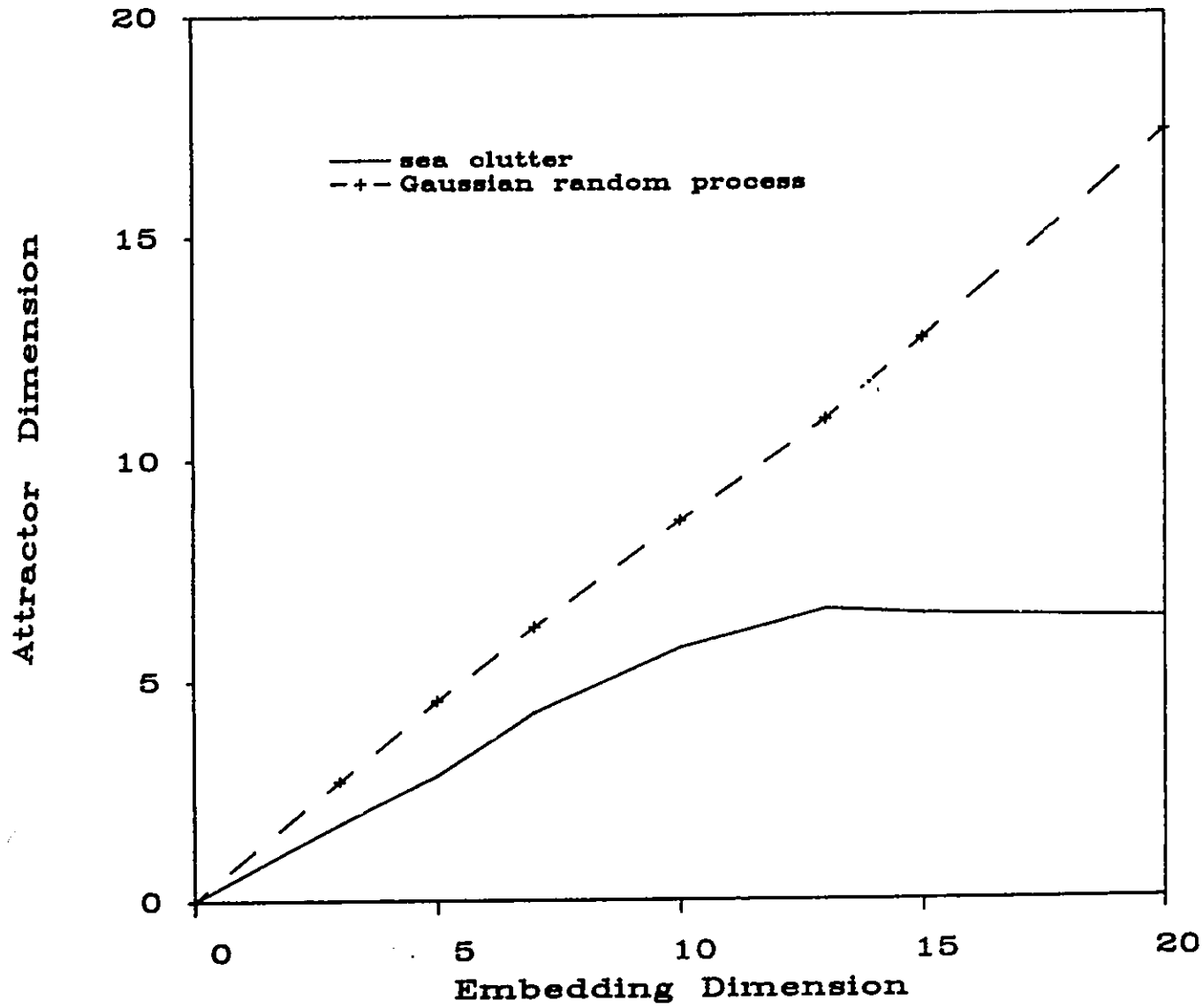


Figure 3.11: Correlation dimension analysis of D1

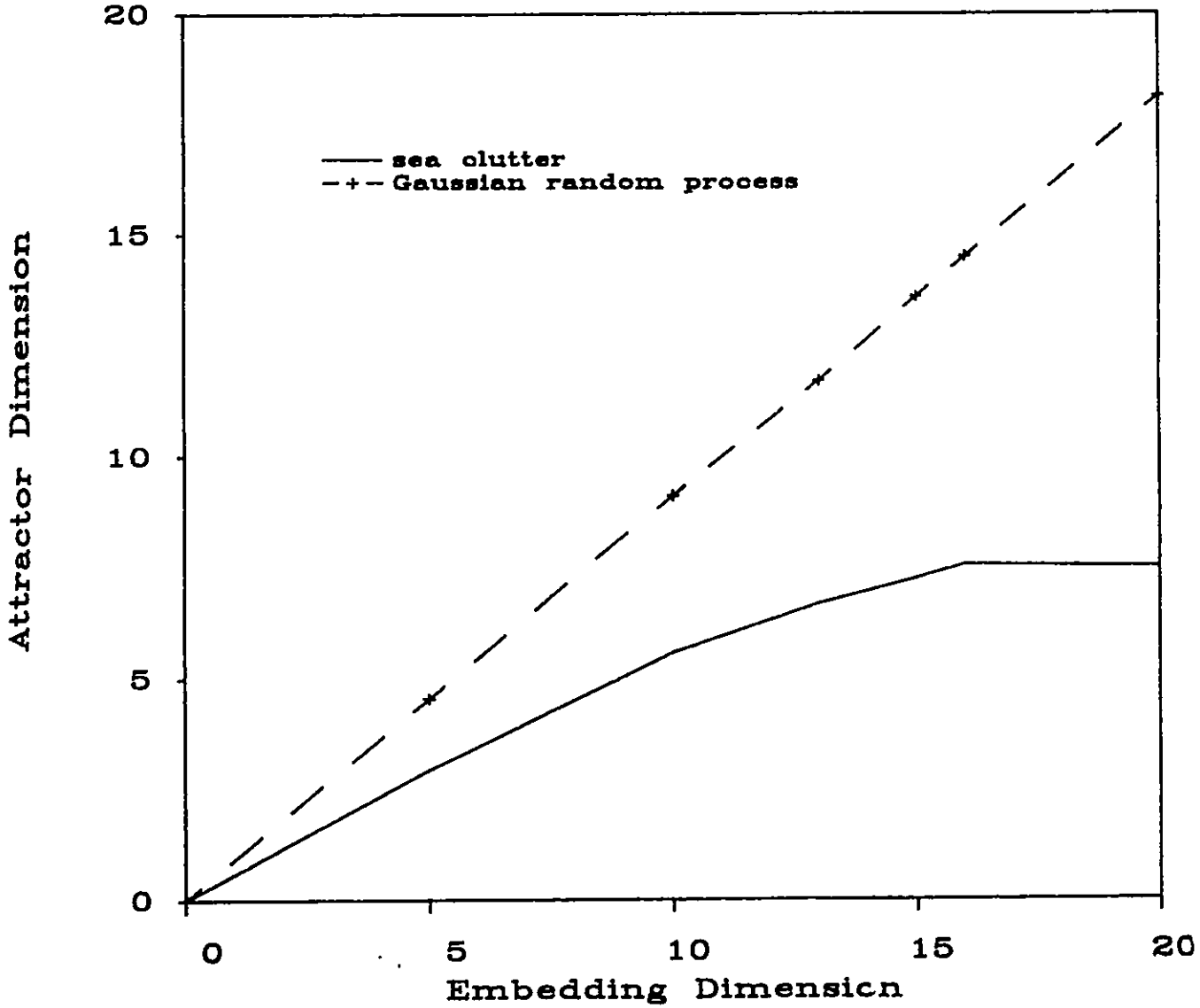


Figure 3.12: Correlation dimension analysis of D2

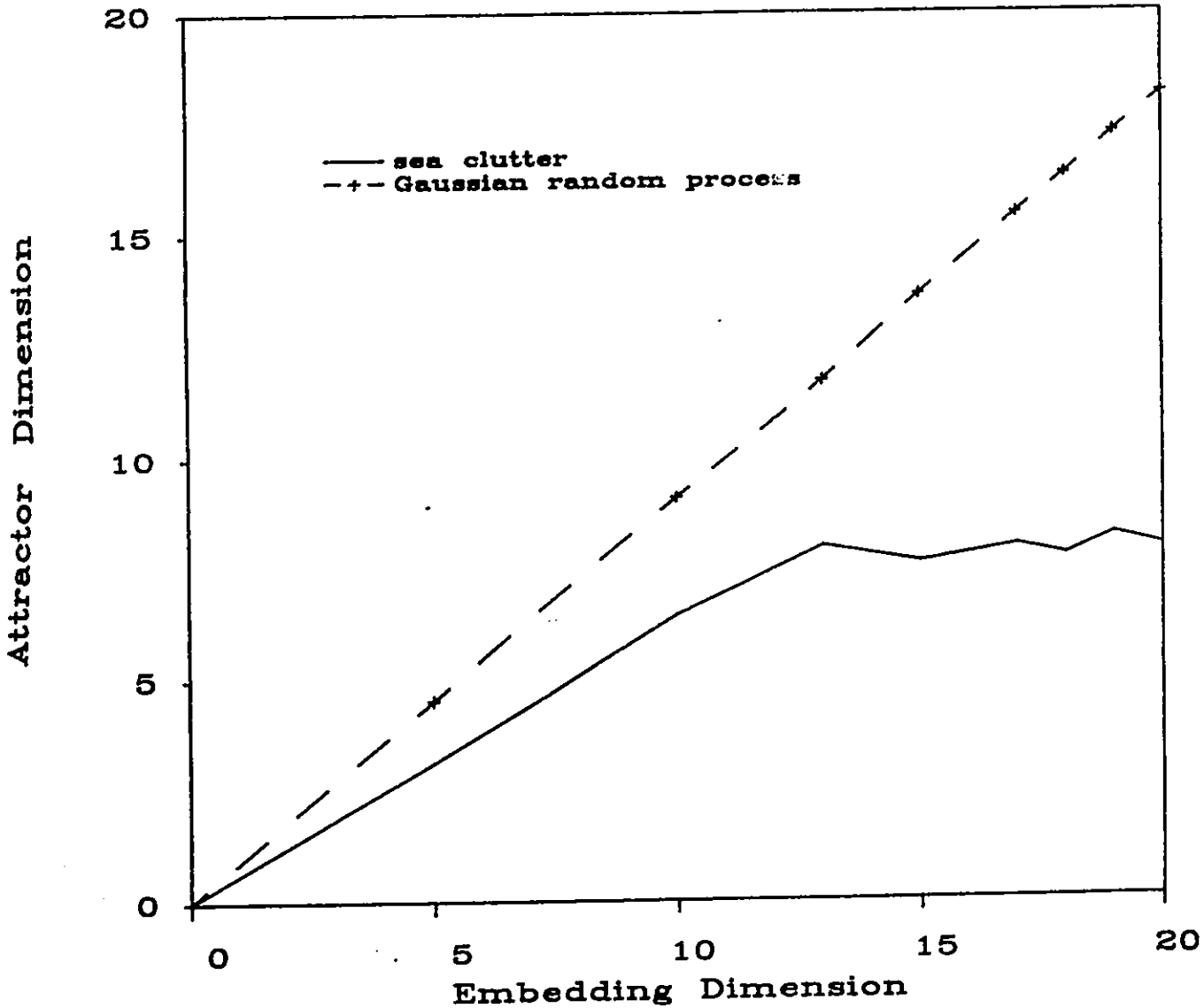


Figure 3.13: Correlation dimension analysis of D3

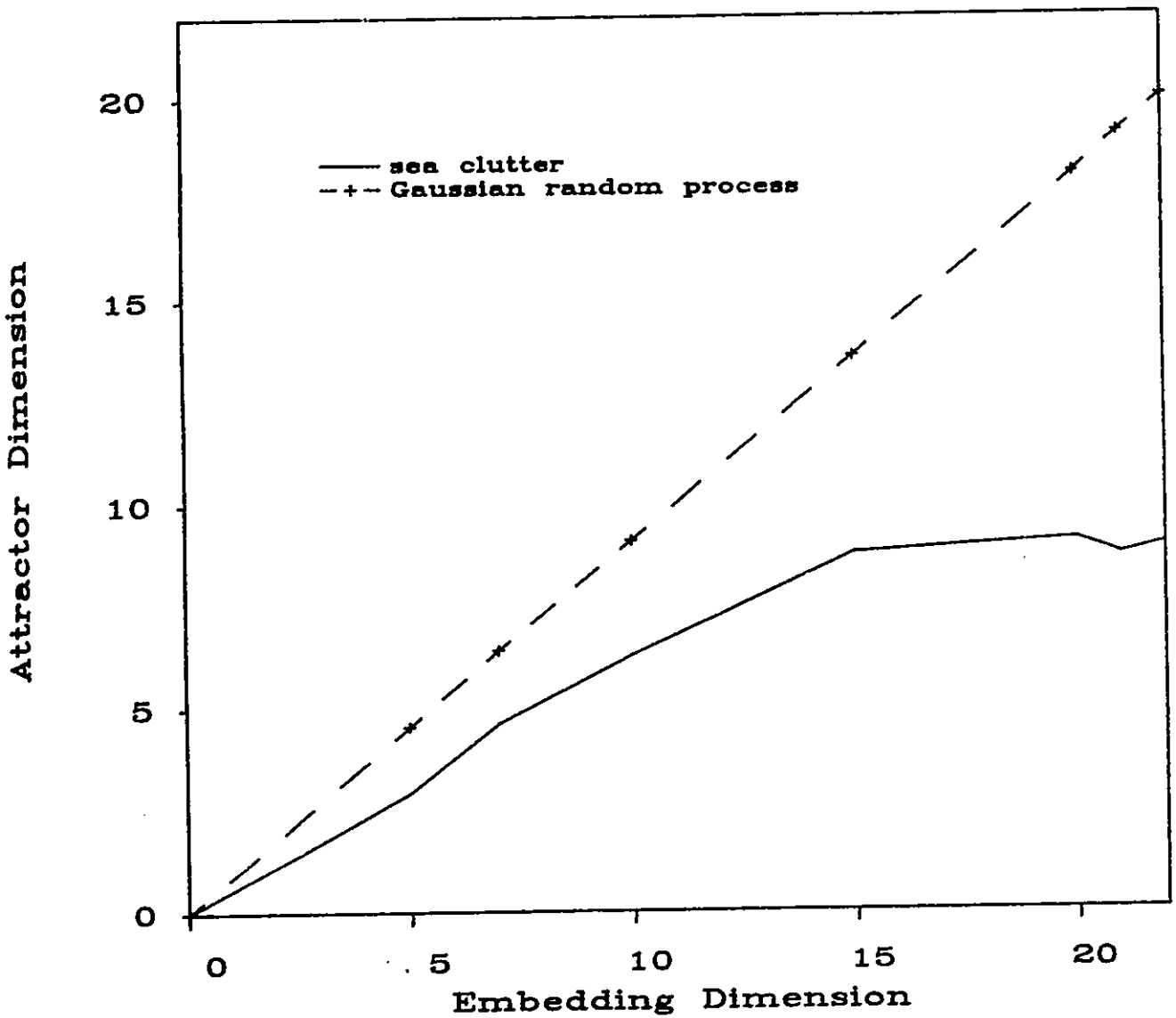


Figure 3.14: Correlation dimension analysis of D4

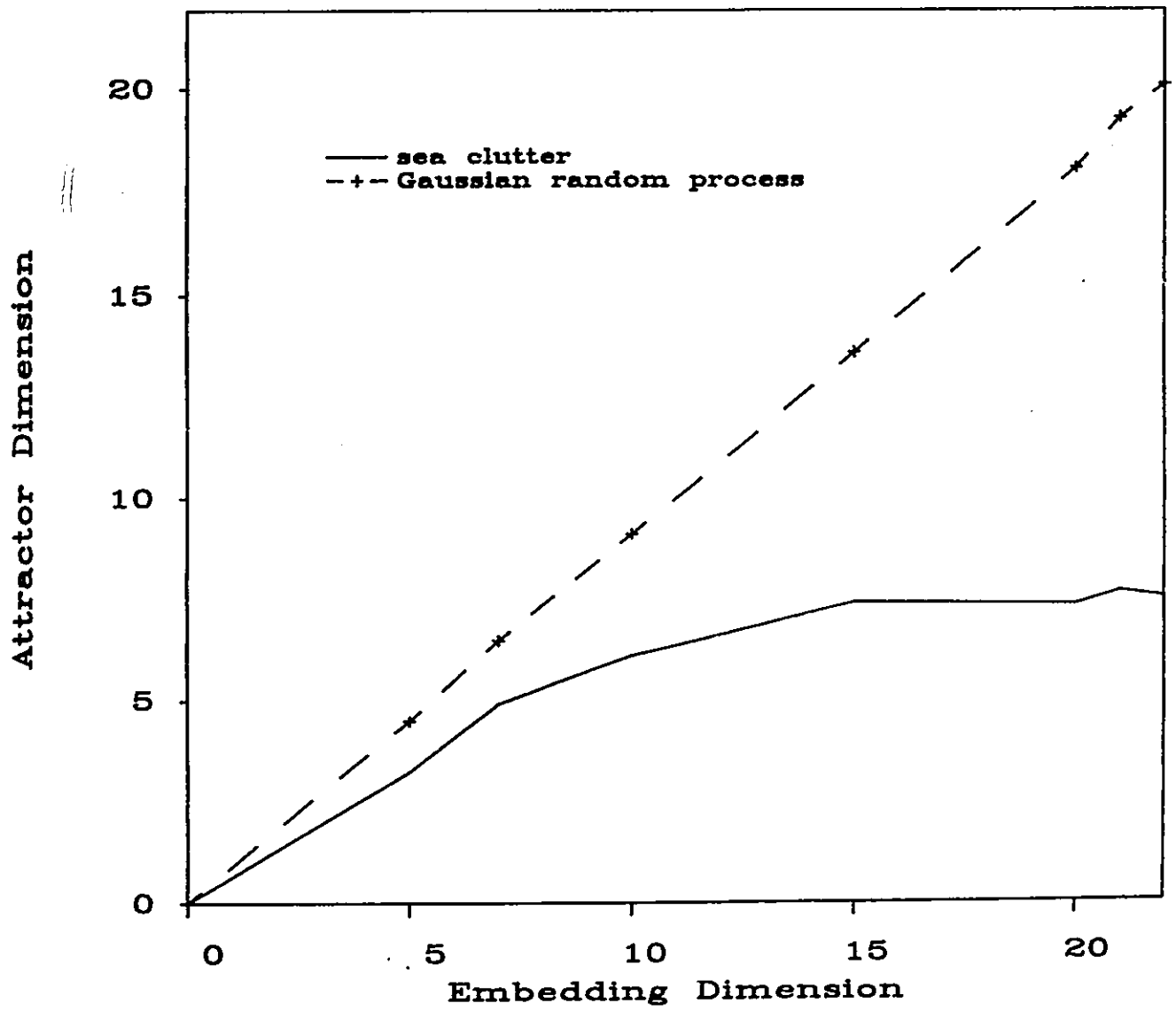


Figure 3.15: Correlation dimension analysis of D5

Chapter 4

Predictability of Sea Clutter on Attractor

4.1 Basic Concepts

As we mentioned in Chapter 1, a crucial point in the study of randomness of a complex system is that of predictability. Predictability is not only a very useful factor in many applications like stock market analysis, weather forecasting, and predictive modeling of sea clutter. It also provides a very clear indication of the existence of chaos. Many people even use some measure of the predictability of a dynamical system in the definition of chaos [41]. Our goal in this chapter is to study the predictability of sea clutter data to further confirm the existence of chaos in sea clutter (we will elaborate on this point later in the chapter). Before studying the predictability of our sea clutter model, we should understand some fundamantal issues about predictability.

Consider two observers looking at a system, observer A sees the system at time t_1 and the observer B at time $t_2 > t_1$. If observer A knows more precisely the state of the system at t_2, the system is said to be predictive. For such a system, earlier

observations convey more information than later ones. In other words, predictions are more accurate than observations. Information is destroyed in a predictive system. If observer B knows more precisely than observer A at t_2, the system is called unpredictive. In an unpredictive system, the later the observation, the more information is gained.

When we apply this idea to a dynamical system, we can say that a predictive system is one where all the trajectories approach one another in the sense that

$$\| \varphi_{t_2}(y) - \varphi_{t_2}(x) \| < \| \varphi_{t_1}(y) - \varphi_{t_1}(x) \| \tag{4.1}$$

for any x and y, $x \neq y$, and any t_1 and t_2, $t_1 < t_2$. In other words, the flows are contracting. On the other hand, an unpredictive system is one where all the trajectories diverge

$$\| \varphi_{t_2}(y) - \varphi_{t_2}(x) \| > \| \varphi_{t_1}(y) - \varphi_{t_1}(x) \| \tag{4.2}$$

for any x and y, $x \neq y$, and any t_1 and t_2, $t_1 < t_2$. Similarly, these flows are called expanding.

Now we see that a deterministic system can be unpredictive if it contains an expanding flow, since the expansion can create information, that is, increase the phase volume. However, we have not addressed the question of how the steady state behavior of a dynamical system can be unpredictive.

To answer this question, let us take a look at a very common and important dynamical structure - the Smale horseshoe. Basically, Smale horseshoe comprises three actions : stretching, squeezing and folding as depicted in Fig. 4.1. The stretching expands the phase space volume of the dynamical system, and hence the dynamical system become unpredictable. Squeezing and folding are just two other operations to keep the phase space in the desired base space. This structure illustrates the fundamental idea of how a deterministic system exhibits unpredictive behavior; namely,

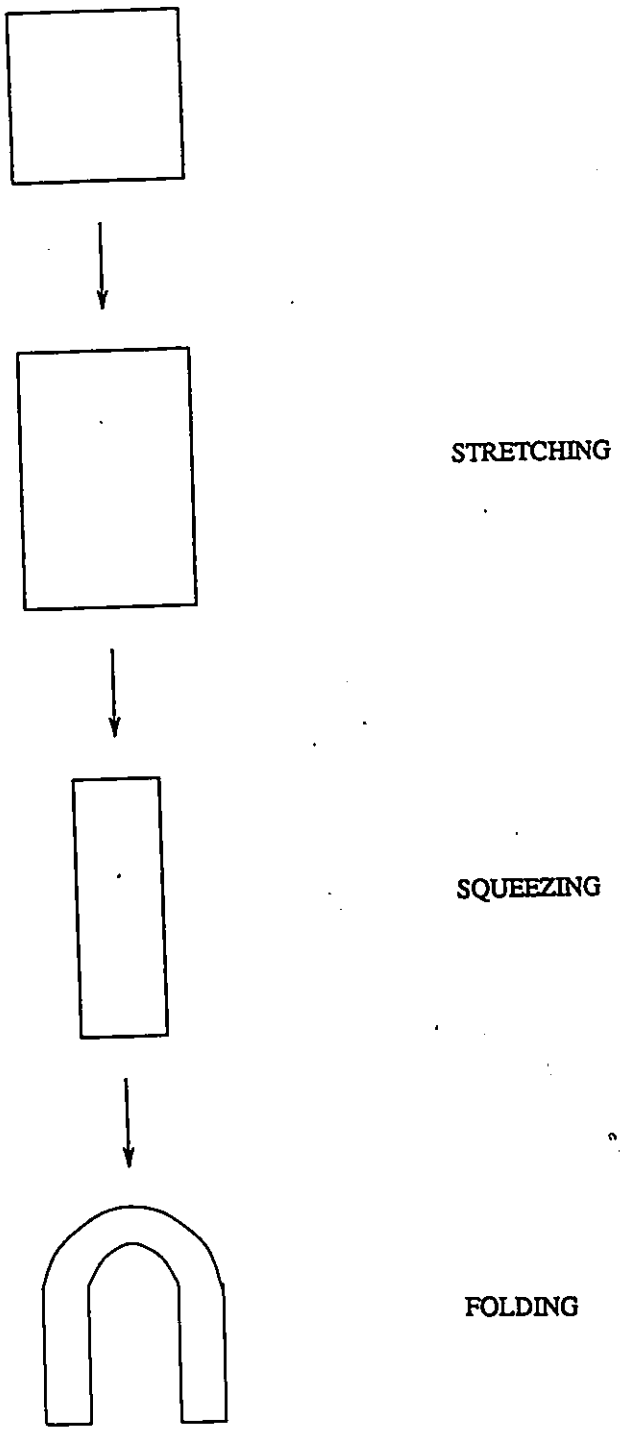


Figure 4.1: Basic structure of Smale horsehoe

the existence of both expanding (stretching) and contracting (squeezing and folding) operations. Although it is nct likely that all unpredictive systems must come from the Smale horseshoe, there is no doubt that the Smale horseshoe does illustrate a very important idea in explaining the unpredictability of deterministic systems.

Based on the discussion so far, we can conclude that the unpredictive behavior of a deterministic dynamical system is due to the expansion of the phase space, that is, increase of information. Therefore, predictability can be defined by the divergence of initially close pieces of trajectories and estimated by the cumulative distance distributions of expanding pairs of points. Consider the difference between two states of a deterministic process at a given time. If this difference is small initially and becomes larger in the future, the deterministic process is unpredictive or unstable. Thus, a relevant measure of predictability is the rate at which initially small errors grow.

Now consider the time evolution of the sea clutter or scattering system. It can be simulated by partial differential equation describing the underlying physical processes. These equations may be conveniently transformed to a set of N ordinary differential equations :

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_N), \qquad i = 1, 2, \ldots, N \tag{4.3}$$

or a set of difference equations :

$$x_i(t+1) = f_i(x_1(t), x_2(t), \ldots, x_N(t)), \qquad i = 1, 2, \ldots, N \tag{4.4}$$

with N suitably normalized variables x_i. In our model reconstruction problem, these equations are constructed from the data and N is then a suitably embedding dimension of the dynamical system. Thus the phase space containing the time evolution is spanned by the N different variables x_i, $i = 1, 2, \ldots, N$. A sea clutter state at an initial time is realized by a vector $\mathbf{x}_0 = (x_1(t_0), x_2(t_0), \ldots, x_N(t_0))$ in phase space. Another realization $\mathbf{x}_0 + \delta\mathbf{x}$ may be defined by an initially small vector or deviation

from the basic state $\mathbf{x_0}$:

$$\delta \mathrm{x} = (\delta x_1, \delta x_2, \ldots, \delta x_N) \tag{4.5}$$

The difference between two states can be measured by the distance $D(t) = (\delta \mathrm{x} \cdot \delta \mathrm{x})^{1/2}$ which evolves as time progresses.

One way to evaluate the predictability is to solve the nonlinear equation Eq.(4.3) twice with slightly different sets of initial conditions. Then $D(t)$ can be evaluated for a sequence of time steps. Beyond the time limit of predictability, $D(t)$ would oscillate about a value not greater than the difference between two randomly selected states of the system. If $D(t)$ stays below this threshold, one can expect predictability for a large time range.

Although this approach works in principle, we find it unfavorable to our problem for at least two reasons. First, the computation of this procedure is too heavy. Second, this approach requires an accurate model Eq.(4.3) for the underlying experimental system. In our problem, what we have is just the experimental data, and the model is just an estimated one. Hence the reliability of evaluating the predictability by an estimated model is questionable.

Another approach is to evaluate the growth rate of error $\delta \mathrm{x}$ in the system, which is governed by the set of differential equations

$$\frac{d\delta x_i}{dt} = \sum_{j=1}^{n} A_{ij} \delta x_j, \qquad i = 1, 2, \ldots, n \tag{4.6}$$

The coefficeints A_{ij} are the elements of the Jacobian matrix of $\mathrm{f} = (f_1, f_2, \ldots, f_n)$, defined by the partial derivative of Eq.(4.3)

$$A_{ij} = \frac{\partial f_i(x_1, x_2, \ldots, x_n)}{\partial x_j} \mid_{x=x_0} \tag{4.7}$$

They are determined at the basic state $\mathrm{x} = \mathbf{x_0}$ changing with each time step; therefore, the elements A_{ij} are, in general, time dependent coefficients which vary with the time evolution $\mathrm{x}(t)$ of Eq.(4.3).

If a state is stable (unstable), $D(t)\ |_{\mathbf{x}=\mathbf{x}_0}$ remains bounded (grows quasi-exponentially) for all time. The local stabilities of the sea clutter evolution are determined by the eigenvalues A_{ij} (or characteristic exponents) of the Jacobian matrix which change with time. If at least one eigenvalue λ has a positive real part, the evolution is unstable and $D(t)$ grows proportionally to $e^{\lambda(t-t_0)}$; otherwise the evolution is stable.

The magnitudes of the positive characteristic exponents can be used as a measure of unpredictability. They define, in a time average sense, the mean rate of divergence of initially close trajectories separated by an infinitesimally small vector $\delta\mathbf{x}$. In this sense, they describe the system's sensitivity on initial conditions. Thus, the intrinsic unpredictability is characterized by the magnitudes of the positive characteristic exponents to be deduced from observations.

The following section introduces the concept of characteristic exponents and how unpredictability of sea clutter can be estimated by the growth rate of infinitesimally small errors using observed data.

4.2 The Characteristic Exponent as a measure of Predictability

First we consider the one-dimensional case. The dynamic is described by a difference equation

$$x(t+1) = f(x(t)) \tag{4.8}$$

and $x(t_0)$ the initial condition generating the reference trajectory $\{x(t)\}$. The stability of this trajectory is determined from the evolution of a neighboring trajectory starting at $\tilde{x}(t_0) = x(t_0) + \delta x(t_0)$. After one iteration, we have

$$\tilde{x}(t_1) = x(t_1) + \delta x(t_1) = f(x(t_0) + \delta x(t_0)) = f(x(t_0)) + f'(x(t_0))\delta x(t_0), \tag{4.9}$$

f' being the derivative of f. The derivative is then given by

$$\delta x(t_1) = f'(x(t_0))\delta x(t_0) \tag{4.10}$$

After a second iteration, by the chain rule of differentiation, we get

$$\delta x(t_2) = f'(x(t_1))\delta x(t_1) = f'(x(t_1))f'(x(t_0))\delta x(t_0), \tag{4.11}$$

and at the nth step

$$\delta x(t_n) = (\prod_{i=0}^{n-1} f'(x(t_i)))\delta x(t_0) \tag{4.12}$$

The evolution of the distance between the two trajectories is obtained after taking the absolute value of this product. Because of an exponential convergence/divergence of the trajectories, we assume $|\delta x(t_n)| \sim (\gamma_{eff})^n |\delta x(t_0)|$, where γ_{eff} is an effective rate per iteration step obtained from

$$\gamma_{eff} = \lim_{n\to\infty} (|\frac{\delta x(t_n)}{\delta x(t_0)}|)^{\frac{1}{n}} = (\prod_{i=0}^{n-1} |f'(x(t_i))|)^{\frac{1}{n}} \tag{4.13}$$

which, once logarithms are taken, gives

$$\lambda = \log(\gamma_{eff}) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \log(|f'(x(t_i))|) \tag{4.14}$$

This limit, called the *Lyapunov exponent*, clearly presents itself as the time average of $\log(|f'|)$, the local divergence rate. After the transient has decayed, all physically relevant trajectories belonging to an attractor are expected to yield equivalent time averages. The so-defined Lyapunov exponent is thus expected to exist and to be independent of the initial condition taken in the basin of attraction of the attractor. Furthermore, it is easily seen to be invariant under smooth changes of variables. Indeed, instead of measuring the state variable itself, we may measure an observable $a = y(x)$, where y is a diffeomorphism. Then, from the time series of the observable a, that is,

$$a(t_{n+1}) = g(a(t_n)) = (y \circ f \circ y^{-1})a(t_n) \tag{4.15}$$

Using $(y^{-1})'(a) = [y'(x)]^{-1}$, by the chain rule of differentiation we obtain easily

$$|\frac{\delta a(t_{n+1})}{\delta a(t_0)}| = y'(x(t_n)) \, (\prod_{i=0}^{n-1} |f'(x(t_i))|) \, (y^{-1})' \, a(t_0) \tag{4.16}$$

or, taking the logarithms as in Eq.(4.14), we have

$$\begin{aligned} \lambda^{(a)} &= \lim_{n\to\infty} [\frac{1}{n} \sum_{i=0}^{n-1} \log(|f'(x(t_n))|)] \\ &\quad + \lim_{n\to\infty} \frac{1}{n} (\log(|y'(x(t_n)|) + \log(|(y^{-1})'(a(t_0))|)) \end{aligned} \tag{4.17}$$

where the "boundary terms" on the second line become negligible when n tends to infinity, so that we have $\lambda^{(a)} = \lambda^{(x)}$; that is, the Lyapunov exponent derived from an observable is the same as the original system.

To generalize the above idea to higher dimensional maps, we consider an N-dimensional map $\mathbf{x}(t_{n+1}) = f(\mathbf{x}(t_n))$. Around a given iterate, we have

$$\delta x_i(t_{n+1}) = \sum_{j=1}^{N} \partial_j f_i(\mathbf{x}(t_n)) \delta x_j(t_n) \tag{4.18}$$

where $\partial_j f_i$ denotes the partial derivative of component f_i with respect to variable x_j. The $N \times N$ matrix $\mathbf{J}(t_n) = [\partial_j f_i(\mathbf{x}(t_n))]$ is by definition the Jacobian matrix of f evaluated at $\mathbf{x}(t_n)$ and, by analogy with Eq.(4.12), in vector notation we have

$$\delta \mathbf{x}(t_n) = (\prod_{i=0}^{n-1} \mathbf{J}(t_i)) \, \delta \mathbf{x}(t_0) \tag{4.19}$$

where the product must remain time-ordered $\mathbf{J}(t_{n-1}), \ldots, \mathbf{J}(t_1), \mathbf{J}(t_0)$ since matrix multiplication is not commutative in general. The stretching of the distance between two neighboring trajectories is given by the average evolution of the length $|\delta \mathbf{x}(t_n)|$ or, preferably by $|\delta \mathbf{x}(t_n)|^2$:

$$\begin{aligned} |\delta \mathbf{x}(t_n)|^2 &= \delta \mathbf{x}^T(t_n) \, \delta \mathbf{x}(t_n) \\ &= [(\prod_{i=0}^{n-1} \mathbf{J}(t_i)) \, \delta \mathbf{x}(t_0)]^T \, [(\prod_{i=0}^{n-1} \mathbf{J}(t_i)) \, \delta \mathbf{x}(t_0)] \\ &= \delta \mathbf{x}^T(t_0) [\mathbf{J}^T(t_0) \mathbf{J}^T(t_1) \ldots \mathbf{J}^T(t_{n-1}) \mathbf{J}(t_{n-1}) \ldots \mathbf{J}(t_1) \mathbf{J}(t_0)] \, \delta \mathbf{x}(t_0) \end{aligned} \tag{4.20}$$

The effective evolution rate is now defined by

$$|\delta x(t_n)|^2 = (\gamma_{eff})^{2n}|\delta \mathrm{x}(t_0)|^2 \tag{4.21}$$

and we have to study the limit

$$\gamma = \lim_{n\to\infty} \sqrt[2n]{\frac{\delta \mathrm{x}^T(t_0)[\mathbf{J}^T(t_0)\ldots\mathbf{J}^T(t_{n-1})\mathbf{J}(t_{n-1})\ldots\mathbf{J}(t_0)]\delta \mathrm{x}(t_0)}{\delta \mathrm{x}^T(t_0)\cdot\delta \mathrm{x}(t_0)}} \tag{4.22}$$

or rather its logarithm $\lambda = \log(\gamma)$.

λ takes one of the N values $\lambda_1, \lambda_2, \ldots, \lambda_N$, which are related to the N eigenvectors of the Jacobian matrix. Thus, the characteristic exponents represent the expansion or contraction of different direction in the phase space. A key remark is that the largest Lyapunov exponent, λ_1, measures the divergence rate of the length of an infinitesimal one-dimensional element in phase space, i.e., the line segment joining two points, one on the reference trajectory, the other on the perturbed trajectory, see Fig. 4.2. Considering two perturbed trajectories with linearly independent initial conditions $\delta \mathrm{x}^{(1)}(t_0)$ and $\delta \mathrm{x}^{(2)}(t_0)$, at any time the two vectors $\delta \mathrm{x}^{(1)}(t_n)$ and $\delta \mathrm{x}^{(2)}(t_n)$ define a parallelogram in tangent space. The evolution rate of the length of these two vectors is still given by the largest Lyapunov exponent λ_1 since they both have a nonvanishing projection onto the most unstable Lyapunov direction. However, besides this "longitudinal" deformation, the parallelogram also experiences a "transverse" deformation at a rate governed by the next-largest Lyapunov exponent. After one iteration, its surface is multiplied by a factor that tends asymptotically to $\gamma_1\gamma_2$, so that, taking logarithms to get the evolution rate ρ_2 of a two-dimensional element, we obtain $\rho_2 = \lambda_1 + \lambda_2$. More generally, for a p-dimensional parallelogram ($p \leq d$, the dimension of space) we have

$$\rho_p = \sum_{i=1}^{p} \lambda_i \tag{4.23}$$

which provides a quantitative measure of predictability; it describes the expansion

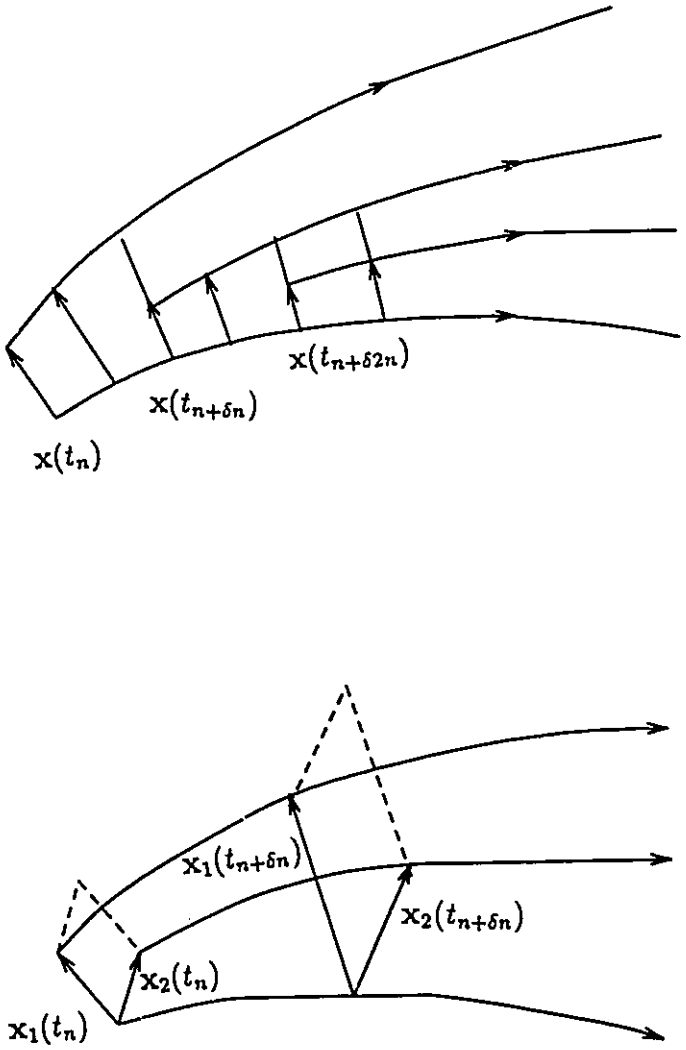


Figure 4.2: Divergence of trajectories

of an infinitesimal ellipsoid to which only the diverging components of the principle axes contribute.

Each positive exponent describes a direction in which the system realizes stretching or divergence, decorrelating nearby states. Therefore, the long-term behavior of an initial condition $\mathbf{x}(t_0)$ cannot be predicted; this characterizes a chaotic system with sensitive dependence on initial conditions. Each positive Lyapunov exponent $\lambda_i > 0$ contributes to the divergence or expansion of a phase space volume element surrounding the initial state $\mathbf{x}(t_0)$, and their sum defines an exponential growth rate of initially small errors.

To summarize the above discussion, traditional predictability analysis determines the error growth from the evolution of an assumed true state disturbed by a random error perturbation. This provides an estimate of the largest Lyapunov exponent. The expansion of an initial sphere of infinitesimal errors growing into an ellipsoid corresponds to all positive Lyapunov exponents, which give more complete information of the dynamical system's sensitive dependence on initial conditions. However, to apply this idea to all sea clutter data, we have to analyze the predictability of the reconstructed phase space dynamics which has an identical Lyapunov spectrum to the original attractor.

Calculation of the largest Lyapunov exponent λ_1 is based upon the definition and the Oseledec Theorem [30] given below

Theorem 4.1 (Oseledec) *Let F be C^2. Let F and its attractor Λ possess an ergodic invariant measure μ. Then there is a μ-measurable set $\Lambda_1 \subseteq \Lambda$ such that $\mu(\Lambda_1) = \mu(\Lambda)$, and such that for all $\mathbf{x} \in \Lambda_1$, Lyapunov exponents exist.*

We briefly explain the Wolf, et al. [41] algorithm which is by far the most reliable, widely used and efficient method for numerically computing Lyapunov exponents.

For each embedding dimension N we use the time series $\{a_i\}$ to form a recon-

structed phase space with a set of vectors $\{\mathbf{a}_i\}$ of N histories. The algorithm is started by locating the nearest neighbor $\mathbf{a}_{i_1} \neq \mathbf{a}_1$ to the initial N-history $\mathbf{a}_1$. Let $d_1^{(1)} = \|\mathbf{a}_{i_1} - \mathbf{a}_1\|$. Note that $d_1^{(1)}$ is the smallest positive distance $\|\mathbf{a}_{i_1} - \mathbf{a}_1\|$. Select a positive integer q and set $d_2^{(1)} = \|\mathbf{a}_{i_1+q} - \mathbf{a}_{1+q}\|$ and store $g_1(q) = \frac{d_2^{(1)}}{d_1^{(1)}}$. We shall call q an evolution time. This ends the first iteration. We are now ready to enter the main program loop.

Ideally, in order to start the second iteration, we would like to find a new N-history $\mathbf{a}_{i_2}$ near $\mathbf{a}_{1+q}$ whose angle $\theta(\mathbf{a}_{i_2} - \mathbf{a}_{1+q}\ ,\ \mathbf{a}_{i_1+q} - \mathbf{a}_{1+q}\ )$ is close to zero. In this way we mimic the definition of Lyapunov exponent given in Eq.(4.17) as closely as possible. The validity of this equation is also guaranteed by the fact that the difference of the direct sums of the eigenspace corresponding to various Lyapunov spectra has full Lebesegue measure.

Motivated by this strategy we choose $i = i_2$ to minimize the penalty function

$$p(\mathbf{a}_i - \mathbf{a}_{1+q},\ \mathbf{a}_{i_1+q} - \mathbf{a}_{1+q}) \ = \ \|\mathbf{a}_i - \mathbf{a}_{1+q}\| \ + \ \hat{w}|\theta(\mathbf{a}_i - \mathbf{a}_{1+q},\ \mathbf{a}_{i_1+q} - \mathbf{a}_{1+q})| \quad (4.24)$$

subject to the nondegenerate requirement $\mathbf{a}_i \neq \mathbf{a}_{1+q}$. Here $\hat{w}$ is a penalty weight on the deviation $|\theta|$ from zero. Store

$$g_2(q) = \frac{d_2^{(2)}}{d_1^{(2)}} \quad (4.25)$$

where

$$d_1^{(2)} = \|\mathbf{a}_{i_2} - \mathbf{a}_{1+q}\|,\ \ d_2^{(2)} = \|\mathbf{a}_{i_2+q} - \mathbf{a}_{1+2q}\| \quad (4.26)$$

This ends iteration two. Continue in this manner. For iteration k, store

$$g_k(q) = \frac{d_2^{(k)}}{d_1^{(k)}} \quad (4.27)$$

where

$$d_1^{(k)} = \|\mathbf{a}_{i_k} - \mathbf{a}_{1+(k-1)q}\|,\ \ d_2^{(k)} = \|\mathbf{a}_{i_k+q} - \mathbf{a}_{1+kq}\| \quad (4.28)$$

and $i = i_k$ minimizes

$$p(\mathbf{a}_i - \mathbf{a}_{1+(k-1)q},\ \mathbf{a}_{i_{k-1}+q} - \mathbf{a}_{1+(k-1)q}) \tag{4.29}$$

subject to $\mathbf{a}_i \neq \mathbf{a}_{1+(k-1)q}$. Continue until k reaches the end of the time series. Set

$$\hat{\lambda}_q = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{log\left(d_2^{(k)}/d_1^{(k)}\right)}{q}\right] \tag{4.30}$$

Thus, through a simple replacement procedure that attempts to preserve orientation and minimize the size of replacement vectors, we have monitored the long term behavior of a simple principal axis vector. Each replacement vector may be evolved until a problem arises, and so on. This leads us to an estimate of λ_1. (Fig. 4.3)

This approach can be extended to as many nonnegative exponents as we care to estimate : $k+1$ points in the reconstructed attractor define a k-volume element whose long-term evolution is possible through a data replacement precedure that attempts to preserve phase space orientation and probe only the small scale structure of the attractor. The growth rate of a k-volume element provides an estimate of the sum of the first k Lyapunov exponents. Since our interest is to measure the unpredictability and confirm the existence of chaos, we will not discuss the estimation of the whole spectrum of Lyapunov exponents anymore. A more detailed explanation can be found in [12].

The procedure above discussed the theoretical idea of estimating λ_1, one major practical implementation problem that is the choice of a suitable evolution time, should also be considered. The distance function or length element is propagated through the attractor for a time short enough so that only small scale attractor is likely to be examined. If the evolution time is too large we may see the distance function shrink as the two trajectories which define it pass through a folding region of the attractor. This would lead to an underestimation of λ_1. We therefore look for a new data point that satisfies two criteria reasonably well; its separation from

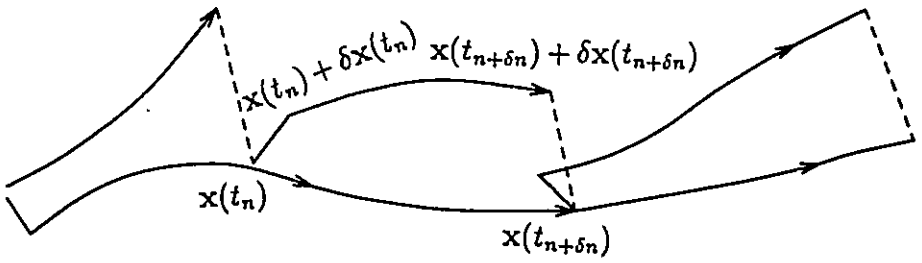


Figure 4.3: Computation of the largest Lyapunov exponent

the evolved point is small, and the angular separation between the evolved and replacement elements is small. In actual computation, a fixed evolution time is usually employed since it would be extremely inefficient to include a varying evolution time.

We have considered the basic theory and computation of Lyapunov exponent, but we have not explained why this computation is important. Not only can this factor measure the unpredictability of sea clutter which is of interest to us, but it is also necessary in determining the existence of chaos.

In distinguishing random process and deterministic chaos, the calculation of the fractal dimension of the attractor which underlies the system evolution in phase space has probably received the widest attention. In fact, most of the studies on chaotic theory in sciences such as economics, physics, atmospheric science and medicine are concentrated on the calculation and explanation of the fractal dimension. Traditionally, a system whose dynamic is governed by a stochastic process is thought to be associated with an infinite fractal dimension in phase space. This is because random noises are generally expected to fill very large dimensional regions of the available phase space. By contrast, finding a finite non-integer value of the dimension is usually considered to be a strong indication of the presence of low-dimensional deterministic chaos. In particular, the precise value of the fractal dimension of the attractor has in addition an important physical significance; the attractor dimension is strictly related to the number of variables needed to describe the dynamics. This is why fractal dimension has drawn a great deal of attention and it is frequently used as an indicator for the presence of strange attractor.

Although the fractal dimension points out the the dimensionality of the experimental data, we feel that it is not sufficient to conclude that a set of experimental data is generated by deterministic chaos solely by a finite non-integer fractal dimenion. The reason is that the fractal dimension is developed from ergodic theory and the computation relies on the existence of invariant measure and the computations

are based on box counting technique or correlation measure. These theories and computations are independent of the ordering of the points in the signal and it is not able to test the differentiablity of the experimental data under study. Whether fractal dimension can distinguish between stochastic and deterministic processes in every situation or not is still an open question. Therefore, it would be wise to put the ordering of the data into consideration; that is, the evolution of the reconstructed vector in the embedded phase space.

We therefore recommend the use of Lyapunov exponent discussed above. In the computation we allow two nearest points in the phase space to evolve and keep on tracking them along the trajectory. This would lead us to a clearer picture of the dynamical behavior occurring in the phase space, because the ordering of the points in the phase space has been taken into consideration. If an exponential divergence of local points is observed, that is, λ_1 is positive, we conclude the existence of deterministic chaos since a random process cannot have a uniform divergent/convergent evolution. If it is in fact the case, the process should not be called random.

Moreover, it is shown recently [10] that two frequency quasiperiodically forced systems can exhibit a new class of dynamical behavior which is termed strange nonchaotic. In this instance, the word strange refers to the geometry of the underlying attractor as exhibiting a fractal structure, while the word nonchaotic refers to the particular dynamics of orbits on the attractor. A chaotic attractor is an attractor for which nearby orbits diverge exponentially in time, displaying sensitive dependence on initial conditions. A nonchaotic attractor, however, is an attractor for which nearby orbits typically do not diverge exponentially in time. Consequently, a strange nonchaotic attractor is an attractor that is geometrically strange, but for which nearby trajectories do not diverge exponentially. This explains why we consider the Lyapunov exponent since we trust that fractal dimension and Lyapunov exponent are the necessary and sufficient measure for the existence of deterministic chaos.

4.3 Local Divergence Analysis

Lyapunov exponents measure the long term average exponential rate of divergence or convergence of initially adjacent phase space trajectories of an attractor, and thus quantify average predictability properties. When at least one Lyapunov exponent is positive, the attractor is chaotic and initially nearby trajectories diverge exponentially on the average. For these attractors, the largest Lyapunov exponent defines a predictability time scale - the average time beyond which deterministic predictions become meaningless owing to the propagation of initial errors over the entire attractor. An increase in the magnitude of the largest Lyapunov exponent implies a decrease in the predictability time scale.

In general, nearby trajectories need not diverge at the same rate on all parts of a chaotic attractor. It has been shown [22] quantitatively for the Lorenz attractor that adjacent trajectories converge in some parts of the phase space, even though these trajectories will separate eventually. This variability of the local divergence rate in the phase space can result from the proximity of trajectories to unstable fixed points and their stable and unstable manifolds. This phase spatial inhomogeneity in the local divergence rate of nearby trajectories is equivalent to a phase spatial variability in the predictability time scale. Clearly, if predictability on a chaotic attractor is a function of time and thus of phase space position, and if short time scales are of interest, then the local divergence rates are the more relevant empirical measures of predictability than the classical Lyapunov exponents. In such cases, an understanding of the temporal and phase spatial variations in the local divergence rate is necessary for a complete quantification of the predictability of the system.

The Lyapunov exponents are not local quantities in either the spatial or temporal sense. Each exponent arises from the average, with respect to the dynamical motion, of the local deformation of various phase space directions. Each is determined by the

long time evolution of a single volume element. Attempts to estimate exponents by averaging local contraction and expansion rates of phase space are likely to fail at the point where these contributions to the exponents are combined. Moreover, the evolution time is not necessarily a constant, and hence the estimation based on a fixed evolution time will introduce errors into the estimation. Other factors including finite data can also be a very serious problem to the estimation procedure. We conclude that Lyapunov exponent calculation by averaging local divergence estimates is a dangerous procedure.

In the computation of the Lyapunov exponent described in the last section, we know that the Lyapunov exponent is just the average of the local divergence rates along a particular trajectory of the dynamical system, and only part of the experimental data is involved. If we obtain a positive Lyapunov exponent, then it must be the result of divergence of local points in the phase space. Thus, evaluating the local divergence can also conclude whether the system is chaotic or not. In other words, local divergence can give us what we want to know by computing the largest Lyapunov exponent. Moreover, there are several advantages in using local divergence instead of the Lyapunov exponent. First, no error is introduced because the evolution time is fixed for each nearby points. As we will show later, the evolution time is quite different for different points. Second, using local divergence can help us to understand the structure of strange attractor. More precisely, knowing the evolution of nearby points on an attractor can give us some information on the topological change of an attractor such as the torsion number. Third, local divergence analysis uses all the given data and hence gives us a better description of the dynamics. Fourth, the computation of Lyapunov exponent is strongly dependent on the stationarity of the experimental data. For slightly nonstationary data which may be an observation of adiabatic dynamical system and hence the attractor may be in fact a union of many attractors (this is very close to the actual sea clutter situation), the estimation

of Lyapunov exponent can produce a completely wrong result. In local divergence analysis, on the other hand, this problem can be handled easily by evaluating the evolution of a neigbourhood of a point rather than just a single point.

Now we perform the local divergence analysis of the sea clutter data. The methodology is quite simple. Assuming that the embedded phase space is constructed (the embedding dimension for each data set is chosen according to its correlation dimension), the experimental data give us a lot of points on the reconstructed manifold. We evaluate each of these points by finding another point in the given set which is very close to it. Then we look at their evolution and evaluate the evolution of the distance between two trajectories.

The result of the analysis are depicted in Tables 4.1 - 4.5. Since there are too many points in the reconstructed phase space, we cannot record the results of all points. We therefore report the local divergence analysis of some points on the attractor only. For data sets D2 and D3, we report points which have a regular time distance of 500 time units; that is, the points correspond to data point on the time series which differ by 500 time units. For D1 and D4, this time step is raised to 1000, and the time step is changed to 2000 for data set D5. The purpose of using points with different time distances is to allow the sea clutter trajectory to have various lengths. For D2 and D3, the attractor's divergence is evaluated in a rather short time period. For D5, the time period is fairly long. Performing analysis on different scales can give us a better understanding of the sea clutter attractor than using a fixed scale. The analysis is performed using an evolution time of 50 time units. This evolution time is sufficient for all the points used in our analysis. We do not record the divergence for all 50 time steps since the behavior becomes random after the distance of two nearby trajectories reaches a maximum value, and this is what we expect.

There are several observations that are quite interesting.

time	x(1000)	x(2000)	x(3000)
1	16.76305	17.66352	26.58147
2	30.13304	31.17691	63.16645
3	45.38722	37.13489	98.81802
4	88.37986	65.04613	110.7475
5	107.3452	99.25724	119.1679
6	126.2181	129.3522	119.5868
7	130.7937	157.5786	132.0492
8	136.5394	169.5612	147.6211
9		184.4017	163.4289
10		192.4266	165.4237
11		189.1904	171.0029
12		212.0259	181.6453
13		258.2305	205.7377
14		296.3157	252.2519
15		321.8929	321.2950
16		337.2343	373.2653
17			379.2453
time	x(4000)	x(5000)	x(6000)
1	16.52271	21.42429	11.13553
2	15.39480	44.45222	29.98333
3	21.28380	81.78019	35.35534
4	26.62705	81.93290	35.80503
5	26.92582	82.80701	48.69292
6	36.06938	85.40491	51.56549
7	46.94678	89.25245	53.55371
8	46.70118	96.28603	53.88877
9	47.38143	103.8027	55.00000
10	47.38143	118.8024	61.52235
11	76.28893	139.3593	66.59579
12	88.09096	152.3844	74.08104
13	116.3400	158.6852	75.63068
14	118.8907	160.5740	
15	123.6528		
time	x(7000)	x(8000)	x(9000)
1	14.83240	11.09054	7.071068
2	24.57641	43.42810	24.69818
3	30.31501	48.02083	24.61707
4	38.97435	47.65501	24.61707
5	58.23229	54.73573	25.70992
6	86.80437	63.39558	26.62705
7	99.67949	70.71068	28.03725
8	103.9760	74.92446	30.54565
9	106.0896	68.09552	31.33688
10	112.4944	70.58328	35.18522
11			39.07685

Table 4.1: Local divergence analysis of D1

time	x(500)	x(1000)	x(1500)
1	17.52142	18.33030	9.797959
2	32.12476	18.97367	12.60952
3	45.35416	35.21363	13.56466
4	62.39391	40.98780	15.26424
5	75.15318	41.07310	14.69694
6	89.26366	42.33202	15.19868
7	95.83319	46.55105	25.92296
8	98.30565	46.76537	33.22649
9	100.1399	48.40454	33.18132
10	103.2327	49.89990	32.75668
11	108.8899		38.05260
12	111.7318		39.44617
13			42.37924
time	x(2000)	x(2500)	x(3000)
1	8.062258	10.14889	6.633250
2	8.062258	12.96148	8.888194
3	8.062258	13.85641	10.58300
4	14.31782	13.85641	13.26650
5	13.45362	13.56466	17.88854
6	15.62050	16.82260	17.97220
7	17.43560	14.79865	20.68816
8	21.07131	14.76482	21.30728
9	25.69047	16.46208	20.54264
10	27.40438	27.47726	25.72936
11	29.37686	32.83291	27.23968
12	27.71281	35.00000	25.84570
13	29.79933	35.00000	34.97142
14	32.86335	35.34119	38.88445
15	41.13393	38.13135	39.05125
16			42.54409
time	x(3500)	x(4000)	x(4500)
1	7.211102	7.280110	8.485281
2	12.80625	11.18034	18.97367
3	12.49000	11.18034	19.87461
4	13.74773	12.24745	20.46949
5	18.13836	12.20656	20.83267
6	20.24846	12.32883	22.22611
7	21.33073	23.76973	23.62202
8	23.13007	26.62705	26.88866
9		31.52777	29.25748
10		33.73426	
11		34.91418	
12		34.91418	
13		47.26521	
14		50.56679	

Table 4.2: Local divergence analysis of D2

time	x(500)	x(1000)	x(1500)
1	9.055386	12.84523	13.85641
2	15.32971	12.84523	14.38750
3	17.66352	27.20294	17.86057
4	19.79899	27.40438	15.09967
5	24.16609	26.98148	20.49390
6	24.26932	26.98148	25.07987
7	27.89265	31.28897	28.08914
8	28.17801	35.87478	33.36166
9	30.57777	35.74913	37.85499
10		33.28663	37.32291
11		36.71512	40.02499
12		38.31449	42.48529
13		40.45986	44.95553
14			46.34652
15			47.50789
time	x(2000)	x(2500)	x(3000)
1	7.615773	7.211102	11.53256
2	14.07125	24.08319	17.11724
3	14.17745	27.83882	20.90454
4	20.02498	29.18904	21.37756
5	23.25941	30.52868	21.54066
6	30.52868	31.04855	21.54066
7	42.09513	31.82766	22.27106
8	43.40507	33.21144	23.38803
9	45.02222	38.75565	24.87971
10			28.53069
11			29.49576
12			32.40370
13			33.37664
14			36.42801
15			40.18706
16			41.71331
17			50.00000
time	x(3500)	x(4000)	x(4500)
1	6.324555	7.615773	6.480741
2	20.02498	11.40175	11.09054
3	20.80865	11.26943	11.26943
4	21.93171	13.67479	15.06652
5	23.15167	17.94436	18.22087
6	23.23790	18.22567	21.61018
7	24.26932	18.24829	22.22611
8	24.35159	21.04757	22.33831
9	27.14774	24.20744	23.13007
10	27.42262	29.54657	25.53429
11			27.20294
12			33.94112
13			33.94112
14			39.28104
15			40.34848

Table 4.3: Local divergence analysis of D3

time	x(1000)	x(2000)	x(3000)
1	8.660254	8.888194	6.633250
2	9.165152	8.888194	6.682763
3	12.12436	21.04757	8.485281
4	22.84732	29.46184	9.219544
5	22.84732	29.34280	9.643651
6	25.65151	29.88311	11.31371
7	26.51415	30.41381	12.20656
8	26.58947	30.80584	12.20656
9	27.47726	33.82307	16.85230
10	29.44486	34.53983	18.65476
11	31.44486		19.57038
12			25.19921
13			26.64582
14			27.14774
15			28.30194
time	x(4000)	x(5000)	x(6000)
1	8.717798	7.071068	8.124039
2	11.87434	7.681146	9.000000
3	16.88194	8.660254	15.23155
4	17.88854	29.29164	15.52417
5	20.02498	30.34798	20.00000
6	29.73214	30.34798	22.80351
7	30.28201	33.00000	24.51530
8	37.37646	36.55133	31.00000
9		37.88139	34.00000
10		39.33192	33.82307
11		39.58535	34.77018
time	x(7000)	x(8000)	x(9000)
1	10.14889	8.717798	6.928203
2	10.19804	9.380832	7.549834
3	10.63015	9.848858	7.071068
4	12.04159	9.848858	21.18962
5	13.00000	12.72792	22.02271
6	16.40122	13.19091	21.93171
7	26.53300	14.45683	22.13594
8	28.70540	18.13836	22.67157
9	29.81610	18.78829	24.16609
10	29.86637	19.20937	24.26932
11	44.00000	22.29350	
12	44.27189	22.89105	
13	46.07173	27.42262	
14		27.58623	

Table 4.4: Local divergence analysis of D4

time	x(1000)	x(3000)	x(5000)
1	9.848858	7.810250	7.874008
2	11.13553	9.000000	7.874008
3	10.90871	10.63015	8.544003
4	12.00000	16.79286	13.60147
5	13.37909	17.14643	19.51922
6	13.26650	19.26136	24.61707
7	24.49490	19.77372	32.29551
8	32.32646	24.14539	43.22037
9	32.37283	24.93993	50.39841
10	38.43176	30.16621	55.82114
11	52.65928	31.79623	55.79427
12	55.20869		57.79427
13	55.93747		59.69087
14	62.93648		68.60758
15			69.93568
time	x(7000)	x(9000)	x(11000)
1	8.000000	6.928203	9.539392
2	10.90871	6.928203	12.40967
3	14.76482	20.19901	17.94436
4	14.66288	21.28380	19.62142
5	16.12452	27.58623	19.39072
6	16.16452	27.73085	25.78759
7	17.08801	27.87472	28.42534
8	17.32051	29.69848	27.78489
9	20.12461	30.23243	37.72267
10	20.19901	39.17908	36.86462
11	22.58318	39.97499	39.52214
12	24.61707		
13	25.86503		
14	26.15339		
15	29.18904		
16	32.26453		
17	32.68027		
time	x(13000)	x(15000)	x(17000)
1	9.110434	5.477226	7.483315
2	9.165152	11.22497	9.165152
3	12.12436	20.34699	14.31782
4	28.67054	21.21320	14.42220
5	29.08608	22.69361	16.37070
6	28.82707	35.25621	18.43909
7	33.98529	36.09709	18.54724
8	34.45287	46.93613	19.10497
9	34.95712	49.58831	20.68816
10	36.95944	50.35872	21.35416
11	37.17526		27.92848
12			38.52272
13			38.32754
14			39.42081
15			42.88356
16			43.43961

Table 4.5: Local divergence analysis of D5

1. All sea clutter data under test were found to have a local divergence behavior. (For convenience of presentation, however, only selected points were included in Tables 4.1-4.5.) This implies that the computation of the largest Lyapunov exponent based on the algorithm described above will give us a positive number. However, we want to stress that this analysis does not give us all the information about the true theoretical Lyapunov exponent value. The local analysis performed here cannot identify whether the divergence is exponential or not. More precisely, we do not observe a smooth exponential divergence behavior in our data. This is because in our experimental study, the data sequence was noisy and of finite length.

2. All sea clutter attractors exhibit variation of predictability on various portions of the attractor. In other words, the rate at which adjacent trajectories diverge on a sea clutter attractor is, in general, not constant, but rather this local divergence rate depends on time and therefore location in the phase space. All five sea clutter data strongly indicate this nonuniform divergence property.

3. The initial distances of the all data sets are pretty close. However, the extents of the divergence are quite different. Comparatively, D1 has the strongest divergence behavior where the divergence value may go up to 380 (x(3000)). D1 also has a richer dynamical behavior compared to the other data sets; x(2000) and x(3000) have a strong divergence and x(8000) and x(9000) diverge in a very slow manner. We can understand that at time instant 9000 or 8000, the trajectory has gone to some part of the attractor which has a weak local divergence behavior.

4. Except for D1, the divergence of the other data sets is not very strong. The reason for this observation is not clear. It may be due to the sea state or some other effects. No conclusion should be drawn by simply using this analysis.

5. Although the divergence of D2 to D5 is not very strong, they all exhibit the nonuniformity of divergence. For instance, x(7000) in D5 takes 17 time units to reach its maximum of 32, and x(9060) and x(11000) take only 10 time steps. Their initial values are comparable, and the maximum divergences are quite close. We may then imagine that x(9000) and x(11000) are very close on the attractor and they have a similar dynamical behavior in this portion, whereas x(7000) though close to x(9000) in time but is in the region of the attractor which has a different dynamical behavior. It is very interesting and useful to partition an attractor into several pieces based on their dynamical properties (Fig. 4.4), since we can predict the behavior of the process by observing its spatial position (see Chapter 6 for a more detailed discussion). However, it seems our understanding about sea clutter attractor is not mature enough to achieve this goal.

This yields a more global picture of how predictability varies on the attractor. Ideally, we would somehow isolate various regions of the phase space that contain portions of the attractor.

In this chapter, we first explain the importance of Lyapunov exponent, and then discuss some of the difficulties in computing this invariant. To understand the divergence behavior of sea clutter, we introduce the use of local divergence analysis. The local divergence analysis is applied to our sea clutter data, and we observe that sea clutter has a local divergence behavior and the evolutions shown correspond to a deterministic (nonrandom) system. Although this observation is not equivalent to knowing the theoretical Lyapunov exponent, local divergence analysis provides another strong evidence that sea clutter should not be considered as a purely random process.

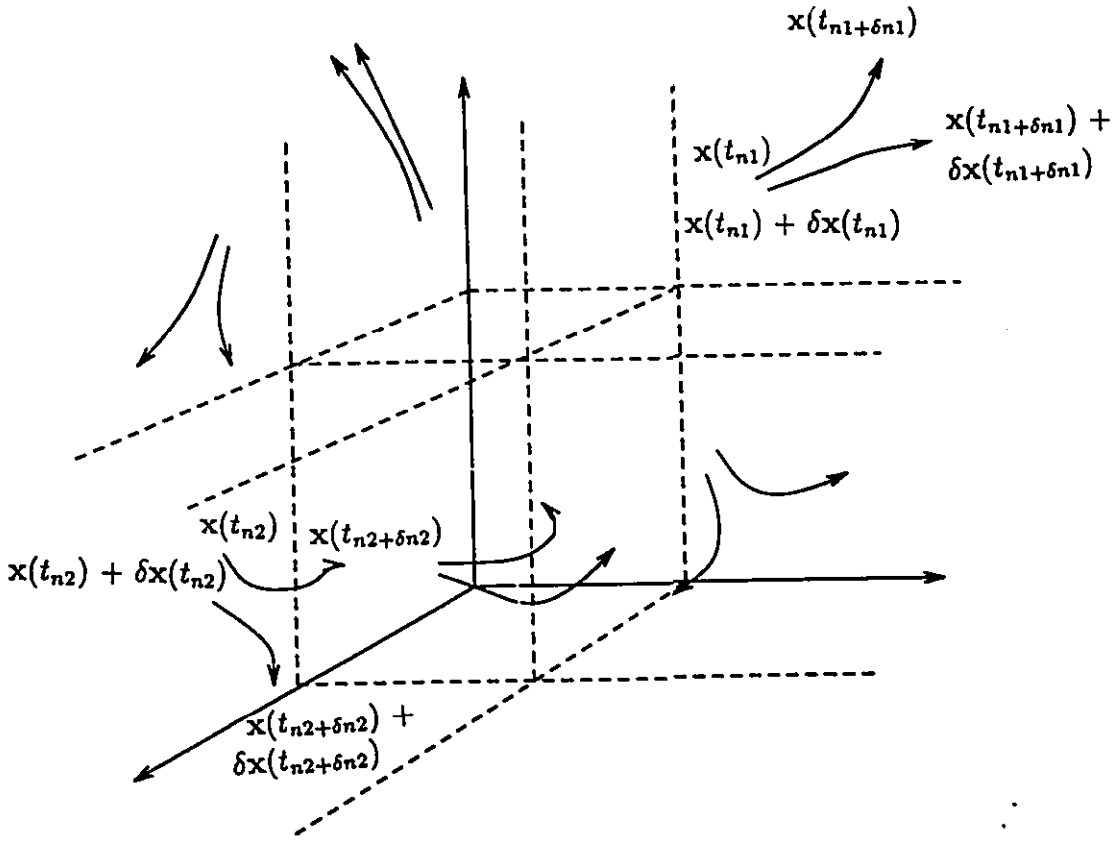


Figure 4.4: Local divergence on various portions of an attractor

Chapter 5

Rational Function Neural Network

5.1 Dynamic Estimation

In the last two chapters, we have demonstrated the potential of modeling sea clutter as a chaotic process. The central problem in applying the theory of dynamical systems is identifying the deterministic, and possibly chaotic, component in a set of observations and distinguishing it from the effects of the ever-present measurement uncertainty, extrinsic noise, and uncontrolled degrees of freedom. This is the inverse problem in nonlinear dynamics : inferring the deterministic equations of motion, if any, underlying observed random behavior in physical systems. Therefore, the next step of our study is to fit a model to the data. There are several approaches available to us. The simplest and most suitable one is to make time discrete and assume that the dynamics can be written as a map in the form

$$\mathbf{x}(n+T) = F(\mathbf{x}(n)) \tag{5.1}$$

where the current state is $\mathbf{x}(n)$, and $\mathbf{x}(n+T)$ is a future state. F and $\mathbf{x}$ are both N-dimensional vectors. The problem is to estimate $\mathbf{x}(n+T)$ as we have discussed in Chapter 2 to reconstruct the dynamics. We will call this estimate $\hat{\mathbf{x}}(n,T)$, and

approximate the dynamics by a map $\hat{F}$ of the form

$$\hat{\mathbf{x}}(n,T) = \hat{F}(\mathbf{x}(n)) \tag{5.2}$$

The random behavior generated by deterministic systems is due to the nonlinear interaction of a small number of degrees of freedom. This raises one of the difficulties in attempting to construct a model of the underlying generator of the observation sequence, since most of contemporary physics and engineering relies on the superposition principle which is a property of linear systems.

The goal in estimating the deterministic equations of motion is to deduce a minimal model that reproduces the behavior. To do this we first need a nonlinear approximating function $\hat{F}$. Second, we need a measure of deviation of the data from a given dynamic F. There are many measures developed for this purpose such as the J-measure, Kullback-Leibler information measure, and mean squares error criterion. Each of these measures has its own advantages and disadvantages, and hence none of them is the best. The one employed here is the mean squares error criterion which is the most popular one. Also, signal processing is a real-time operated problem; thus we must consider an estimation process that can be operated in real time. The conventional linear adaptive filter is one such example. In that sense, a highly complex, nonadaptive dynamic estimation method should be neglected in our application even though it may be very accurate.

Therefore, we attempt to construct the model by learning from the environment. Our goal is to detect and model deterministic structure in noisy data. This is exactly where learning theory can contribute since one of its mandates is to formalize learning paradigms. Computational learning theory has formalized a range of learning paradigms for inductive inference. In this it provides a language and a collection of complexity theoretic methods appropriate to the inverse problem. When integrated with the geometric and statistical techniques of dynamical systems theory, the result

is a framework for consistently distinguishing between deterministic chaotic behavior and extrinsic information sources to which it is coupled.

To specify the inverse problem as a problem in inductive inference, several components need to be defined. The basic components of an inductive inference problem are a rule space, a hypothesis space, an example set, an inference method, and its success criterion. The rule is the underlying dynamical mapping F. The hypothesis space is the space of noisy discrete-time dynamical systems. The example is the set of examples for learning, that is, the observations. The inference method is the model reconstructed from the data, that is, $\hat{F}$. The success is a criterion to measure the goodness of fit. We now see that the inverse problem of chaotic modeling can be formulated as an inductive inference problem in learning theory. The general question is then, given a set of noisy observations, how to infer that some portion of the noise is due to the fluctuating force and how much is due to the deterministic chaos.

The classical procedure for estimating the dynamics uses Bayesian statistical inference to estimate nonlinear models from reconstructed chaotic data series. When approached from this point of view, the estimation of dynamics can be reduced to the model-fitting problem suggested at the beginning of this chapter. This type of learning problem found in dynamical systems is of a different character than the symbolic AI problem traditional in learning theory. They are more akin to problems in the area of neural networks.

Recent developments in artificial neural network have drawn the attention of many researchers. Of all the practical information processing operations that neural networks can currently carry out, one of the most useful is the ability to learn a mathematical mapping by adaptation in response to examples of the mapping's action. This property is important since it has applications in many areas such as function approximation, data compression, speech and pattern recognition. A very important class of applications is nonlinear signal processing, particularly the prediction problem

for a chaotic time series. In this case a network learns input sequences and produces an approximation to an unknown mapping for a deterministic system on its attractor. Such systems may tell the difference between purely random and deterministic processes, and in the latter case allow longer time prediction.

Typically, we are presented with a set of examples $(\mathbf{x}(t_1), \mathbf{y}(t_1)), (\mathbf{x}(t_2), \mathbf{y}(t_2)), \ldots$ of the action of a function $f : R^m \rightarrow R^n$, where $\mathbf{y}(t_k) = f(\mathbf{x}(t_k))$. The requirement is to design a neural network that implements an approximation to the function f; such a network is called a *mapping neural network.*

The most commonly used mapping neural networks employ an underlying model and adjust the weights of the networks to retrieve the useful information from the inputs. The backpropagation network and the radial basis function network are two examples. The backpropagation network is usually implemented by employing multiple layers of sigmoidal nonlinearity [25] as shown in Fig. 5.1. The radial basis function network has several versions : fixed centers [8] (Fig. 5.2), nonlinear [8] (Fig. 5.3), and hybrid [30], see Fig. 5.4.

The backpropagation network, is by far the most popular neural network in use today. However, the backpropagation network suffers from many shortcomings. For instance, the issue of convergence, that of picking a suitable structure, and the issue of obtaining a global minimum are all important factors in designing a useful neural network. Unfortunately, the analytic resolution of these issues in the context of the backpropagation network is still lacking. But, the most serious limitation of the backpropagation network is its inefficiency. More precisely, a backpropagation network has a very slow rate of convergence, and it requires repeated presentation of the training data. While this property may make sense for solving "off-line" problems, it is usually unacceptable for solving real-time problems found in such areas as adaptive signal processing or biological information processing.

Researchers have recently begun to re-examine the method of radial basis functions

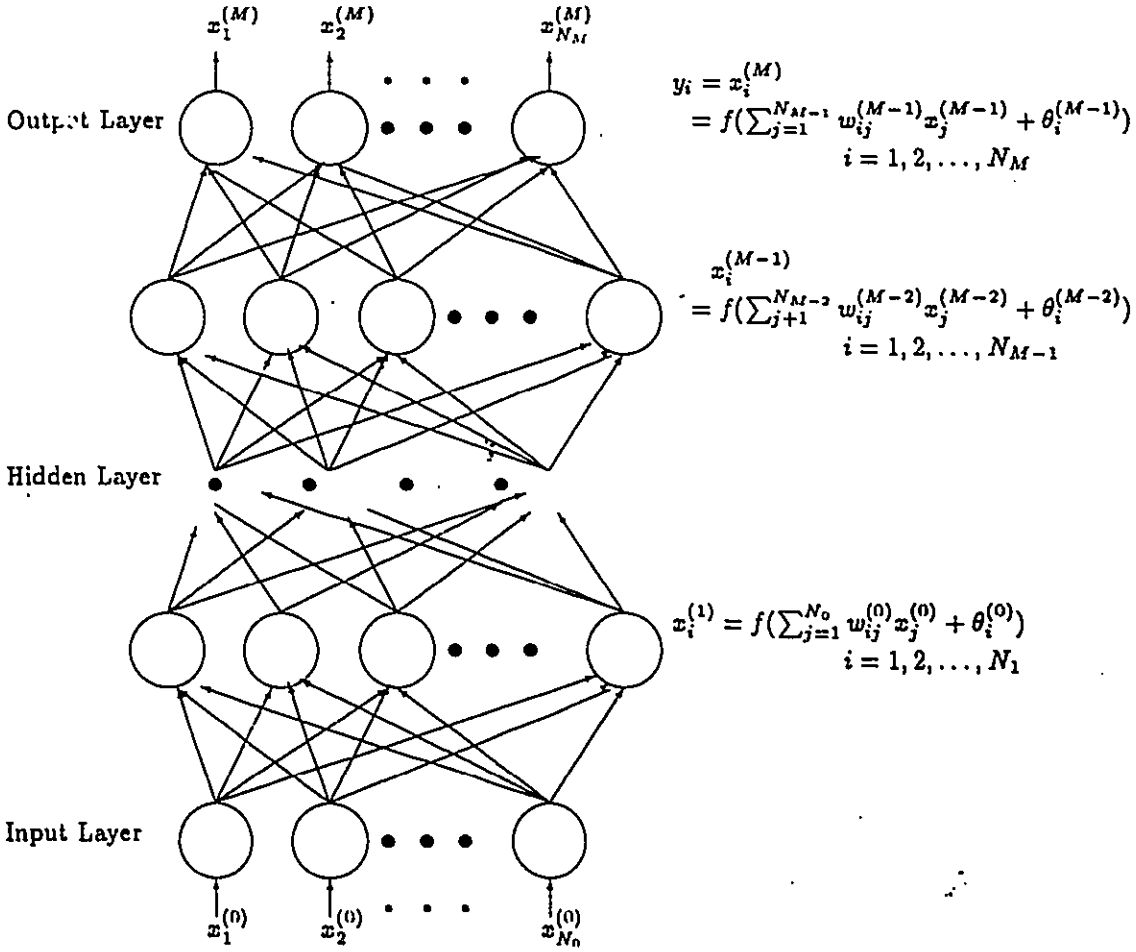


Figure 5.1: Multilayer backpropagation neural network

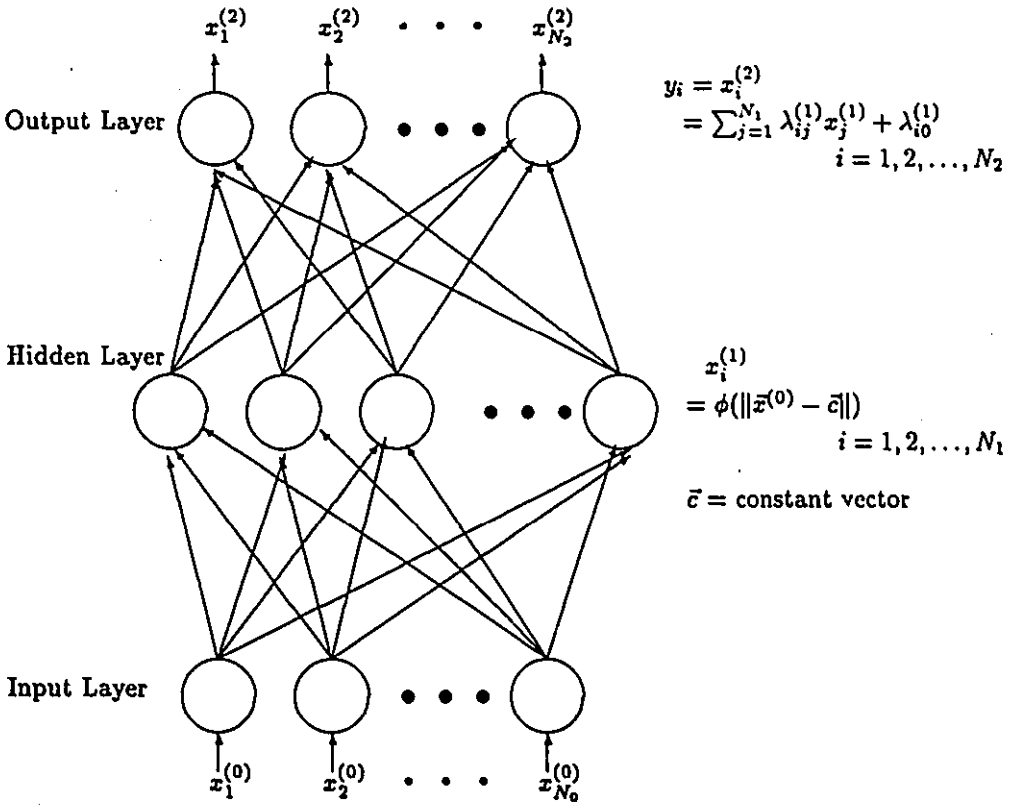


Figure 5.2: Fixed center radial basis function neural network

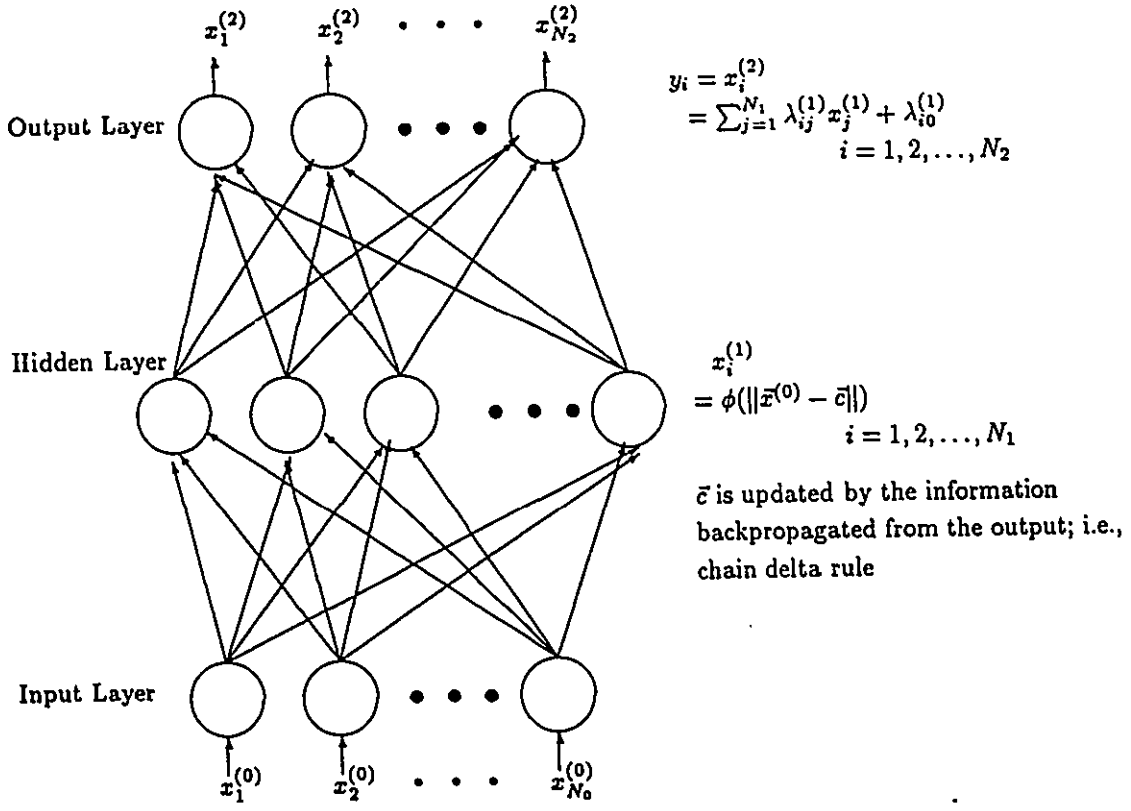


Figure 5.3: Nonlinear radial basis function neural network

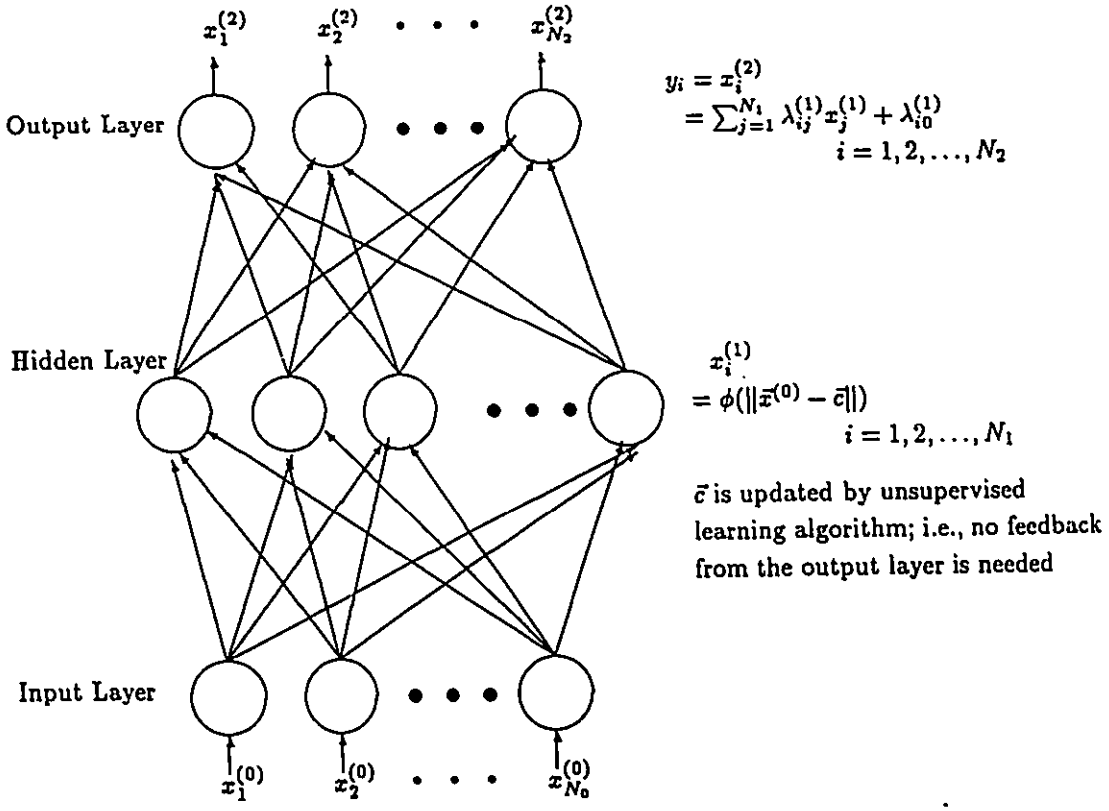


Figure 5.4: Hybrid radial basis function neural network

to build a more efficient neural network. The radial basis function approach constructs a linear function space that depends on the position of the known data according to an arbitrary distance measure. Hence, it has the potential of achieving an efficient learning process. Also, a radial basis function provides a technique for interpolation in high dimensional space. As pointed out in [34], the generalization ability of a neural network is strongly related to the interpolation property of the approximation method used in its design. Unfortunately, a radial basis function is nonlinear in the centers' parameters. Although these parameters can be determined by gradient optimization as in the backpropagation network, and this approach gives a slightly better performance than the backpropagation network [30], it suffers from the same problem as the backpropagation network does. One way to handle this problem is to fix the centers either randomly or by the use of some prior information if available [8]. In such a case, the convergence rate is greatly improved since the function is totally linear in its parameters. However, the performance obtained in this way is worse than that of backpropagation network, in general. Indeed, this new network is not a universal approximator any more as the backpropagation [23] or nonlinear radial basis function network. The behavior of a radial basis function network with fixed centers can be improved by choosing more centers (at least in principle). However, the price to pay here is that of a slow learning process due to the large number of parameters (tap-weights) used in the network. This raises an interesting open problem, namely, that of choosing a suitable number and positions of the centers. Recently, it has been suggested that the positions of the centers can be estimated by using the k-means algorithm, and the combination forms a hybrid structure. This appears to be a good idea, but such a hybrid network needs not only lots of examples to estimate good positions for the centers but also a large number of hidden units so as to achieve a good result, as reported in [30]. More seriously, however, the learning is quite slow because of the k-means algorithm (k- means involves a lot of distance

computations and comparisons). Nevertheless, the network can still perform learning without repeated training.

We would therefore like to devise a neural network such that its parameters can be fitted in a linear least squares fashion, and yet it is still a universal approximator. The Kolmogorov-Gabor polynomial is a good representation that car satisfy these requirements. Indeed, neural networks using polynomials have recently attracted attention [4]. However, polynomials have the disadvantage that there are too many parameters to be determined when the degrees of the polynomial is high. Also, polynomial networks do not extrapolate very well beyond their domain of vaildity, since the norm of a polynomial approached infinity as the norm of the variables tends to infinity. The implication of the second point is that polynomial neural networks may have a poor generalization ability.

In this thesis we propose a network architecture that uses a rational function [27] to construct a new mapping neural network. Here we recognize that rational approximations extrapolate better than polynomials. This is particularly so when the numerator and denominator are of the same degree, since they remain bounded as the norm of the variables approaches infinity. Besides, rational functions contain polynomials as a subset and hence retains some modeling advantages of using polynomials. For instance, in optics, it has been shown [33] that the quadratic term describes many optical transformations. This is, in fact, due to the inherent bilinearity in optics, which results from the quadratic relation between the optical intensity and the optical field. Thus, the polynomials method works well in image enhancement, edge detection, and interpolation of TV image sequences. We can also expect that rational functions will work well in these cases, while other networks such as the backpropagation network may not be able to perform as well as them. Rational functions can also reduce the complexity problem of polynomials. Furthermore, like polynomials, the parameters (tap-weights) can be estimated by a linear adaptive algorihm such

as the Least Mean Squares (LMS) algorithm or the Recursive Least Squares (RLS) algorithm. We consider this property as a major advantage of using rational functions, since it is the only well-known nonlinear function approximation technique to the best of our knowledge that can be trained linearly without any modification of the function.

5.2 Rational Function Approximation

Function approximation by rational functions was first considered by Chebyshev, although rational interpolation was considered by other mathematicians earlier. This nonlinear approximation technique is still one of the most important techniques in use today. We usually think of the representation of special functions by rational approximation for use in a computer. However, the use of rational function approximation is more profound. In particular, rational approximation arises quite naturally in the numerical solution of ordinary and parabolic differential equation, and in the study of other numerical methods. Furthermore, the Stieltjes and the Hamburger moment problem can be well understood via methods based on rational approximation. These applications confirm the significance of rational functions in approximation theory, and therefore their suitability in the design of a mapping neural network.

A rational function $\mathcal{R} : R^m \rightarrow R$ is a quotient of two Kolmogorov-Gabor polynomials, as shown by

$$y = \frac{a_0 + \sum_{j=1}^{m} a_{1j}x_j + \sum_{j=1}^{m}\sum_{k=1}^{m} a_{2jk}x_j x_k + \cdots}{b_0 + \sum_{j=1}^{m} b_{1j}x_j + \sum_{j=1}^{m}\sum_{k=1}^{m} b_{2jk}x_j x_k + \cdots} \tag{5.3}$$

where $x_1, x_2, \ldots, x_m$ are the scalar input to the system. The set $(x_1, x_2, \ldots, x_m)$ forms a vector in R^m, and y is the value of the mapping of that vector in the range R. The representation Eq.(5.3) is unique, up to constant factors of the numerator and the denominator polynomials. The rational function must clearly have a finite order for

it to be useful in solving real-life problems. If the order of the numerator polynomial is α, and the order of the denominator polynomial is β, then we say that the rational function has order (α, β), and so denote it by $\mathcal{R}_{\alpha\beta}$.

Each continuous function may be approximated by a rational function; this is an immediate corollary of the *Theorem of Weierstrass*. Its generalization to multivariate functions then asserts that any continuous multivariate function can be uniformly approximated by a rational function of sufficiently large degree [27]. Therefore, if our neural network is constructed to implement a rational function, it should retain this important property. We can summarize this important observation in the following theorem :

Theorem 5.1 (Network Approximation Theorem) *Rational function neural networks are dense in the space of contiuous functions on compact sets of Euclidean space, in the sense that for any continuous function f, there exists a sequence of rational network functions $\mathcal{R}_n$ that converges uniformly to f.*

Proof: We consider the Banach space $C(K)$ whose members are the continuous functions on K. Let $\mathcal{R}$ be the subspace of $C(K)$ which consists of the restriction to K of those rational functions. We claim that the closure of $\mathcal{R}$, $\overline{\mathcal{R}}$, is all of $C(K)$.

By the Hahn-Banach Theorem [35], there is a bounded linear functional on $C(K)$, say L, which vanishes on $\mathcal{R}$ and $\overline{\mathcal{R}}$ and is not identically equal to zero. The Riesz Representation Theorem [35] then tells us that this bounded linear functional L is of the form

$$L(f) = \int_K f \, d\mu \tag{5.4}$$

where μ is a Borel measure on K, f is any function belonging to $C(K)$. Thus what we need to prove is the following problem : If μ is a Borel measure on K such that

$$\int_K R \, d\mu \; = \; 0 \tag{5.5}$$

for every rational function R, and if f belongs to $C(K)$, then we also have

$$\int_K f \, d\mu = 0 \tag{5.6}$$

Instead of considering an arbitrary compact set K, it is sufficient to consider a square. The extension is guaranteed by the Tietze theorem [27]. For a N-dimensional Euclidean space, we can then choose a Nth-order Bernstein polynomial [27] as an approximation to f. Eq.(5.6) is valid and since polynomials are dense in $C(K)$, $\mathcal{R}$ is dense in $C(K)$.

Now that we know a rational function neural network approximation is possible, we are confronted with the following questions :

1. Is there a rational network $\hat{\mathcal{R}} \in \mathcal{R}_{\alpha\beta}$ which is the best approximation to some specified function f ?
2. Can we say anything about how fast the approximation error tends to zero ?
3. How can we calculate $\hat{\mathcal{R}}$?

These fundamental questions about rational function approximation were actually studied around the turn of the century; however, not all of them have been solved completely, at least to the best of our knowledge.

The first question was answered a long time ago. Although the original proof given by Walsh [39] assumed that the base space was a compact real interval, from the abstract viewpoint of Banach space theory, the extension to multivariate approximation does not introduce any serious difficulty to this particular existence problem. The existence of best approximation is a very important problem in function approximation. A model which has this property is usually considered to be superior to those without this property. For instance, the conventional multilayer backpropagation network is one which does not have the existence of best approximation property

though it can approximate any continuous functions. This fact can be illustrated by a simple example. The functions $f_i(x) = \frac{1}{i}(\frac{1}{1+exp(-[w\cdot x+\theta])} - \frac{1}{1+exp(-[w\cdot x+\theta+i])})$ belong to the class of sigmoidal functions. However, the limit function as i approaches infinity is obviously not a member of sigmoidal functions. Hence, the set is not a closed set and hence it is not possible for the backpropagation network to have the existence of best approximation property. Question 2, relating to the degree of rational approximation or the speed of approximation, can only be determined in some rare cases like the exponential function, absolute function, ... etc. A general and explicit formula is still lacking, although some partial solutions to this problem exist [27]. We feel that they are not well developed yet for giving us the information on how fast the rational function approximations with different orders converge.

As for question 3 pertaining to the composition of the best approximation, there are certainly many techniques in the literature of function approximation theory to accomplish this job. However, these methods are not applicable to our problem, since our objective is to build a network that can learn the underlying environment adaptively. Any "off-line" technique is unfavourable. Therefore, we would like to apply the adaptive estimation technique to compute the rational function $\mathcal{R}_{\alpha\beta}$, which is closest to the desired function.

Assume that we have an (α, β)-rational function and the desired function is d. To get the best approximation means that a rational function belonging to $\mathcal{R}_{\alpha\beta}$ is sought so as to minimize the following integral :

$$min \int [d(x) - \mathcal{R}_{\alpha\beta}(x)]^2 \, d\mu(x) \tag{5.7}$$

over some region. Since continuous waveforms of the functions are usually not available, we can interpret the above minimum mean square error criterion in discrete form :

$$min \sum_{i=1}^{t} |d(i) - \mathcal{R}_{\alpha\beta}(i)|^2 \tag{5.8}$$

where t is the total number of examples available for learning.

Since all components inside the summation sign are positive, the criterion given in Eq.(5.8) can be written as

$$min \ \ |d(i) - \mathcal{R}_{\alpha\beta}(i)|^2 \qquad i = 1, 2, \ldots, t \tag{5.9}$$

Substituting Eq.(5.3) into Eq.(5.8), we have

$$min \ \ \left| d(i) - \frac{N(i)}{D(i)} \right| \qquad i = 1, 2, \ldots, t \tag{5.10}$$

where

$$N(i) = a_0 + \sum_{j=1}^{m} a_{1j} x_j(i) + \sum_{j=1}^{m} \sum_{k=1}^{m} a_{2jk} x_j(i) x_k(i) + \ldots + \sum_{j=1}^{m} \cdots \sum_{p=1}^{m} a_{\alpha j \ldots p} x_j(i) \ldots x_p(i)$$

$$D(i) = b_0 + \sum_{j=1}^{m} b_{1j} x_j(i) + \sum_{j=1}^{m} \sum_{k=1}^{m} b_{2jk} x_j(i) x_k(i) + \ldots + \sum_{j=1}^{m} \cdots \sum_{p=1}^{m} b_{\beta j \ldots p} x_j(i) \ldots x_p(i)$$

The estimation problem is to find the coefficients $\{a_j\}$ and $\{b_j\}$ so as to achieve the minimum.

If $\mathcal{R}_{\alpha\beta}(i)$ in Eq.(5.9) is replaced by a linear function, then we have the case of a single neuron, and the LMS and RLS algorithm can be used to perform the optimization. Returning to Eq.(5.10), we see that the rational function does not provide a linear approximation, that is, the function is nonlinear in the parameters of the numerator and denominator polynomials. In general, to estimate the parameters of the function, a nonlinear learning procedure should be used. Fortunately, when we take a closer look at Eq.(5.10), we observe that Eq.(5.10) is equivalent to the following problem

$$min \ \ |d(i)D(i) - N(i)|^2 \qquad i = 1, 2, \ldots, t \tag{5.11}$$

Eq.(5.11) is valid because a rational function is unique up to a constant. Substituting the expressions of $D(i)$ and $N(i)$ into the objective function in Eq.(5.11), we have

$$min \ |d(i)\{b_0 + \ldots \qquad + \textstyle\sum_{j=1}^{m} \cdots \sum_{p=1}^{m} b_{\beta j \ldots p} x_j(i) \ldots x_p(i)\}$$

$$-a_0 - \ldots - \sum_{j=1}^{m} \cdots \sum_{p=1}^{m} a_{\alpha j \ldots p} x_j(i) \ldots x_p(i)|^2 \qquad (5.12)$$

$$i = 1, \ldots, t$$

Without loss of generality, b_0 can be assumed to be unity, and so Eq.(5.12) becomes

$$min\ |d(i) - \{a_0 + \ \ldots + \sum_{j=1}^{m} \cdots \sum_{p=1}^{m} a_{\alpha j \ldots p} x_j(i) \ldots x_p(i) - \sum_{j=1}^{m} b_{ij} d(i) x_j(i) - \ldots$$
$$- \sum_{j=1}^{m} \cdots \sum_{p=1}^{m} b_{\beta j \ldots p} d(i) x_j(i) \ldots x_p(i)\}|^2;\ i = 1, \ldots, t \quad (5.13)$$

Since the quantity inside the brackets is linear in the parameters, we can express Eq.(5.13) using matrix notations as follows

$$min\ \ |[TARGET] - [DATA][WEIGHT]| \qquad (5.14)$$

where $[DATA]$ is the matrix consisting of all the $d(i)$ and $x_j(i)$ combinations in Eq.(5.14), and $[WEIGHT]$ is the column vector consisting of all the unknown parameters a_j and b_j. In particular, we have

$$[DATA] = \begin{pmatrix} 1 & x_1(1) & \ldots & x_1^2(1) & x_1(1)x_2(1) & \ldots & x_m^\alpha(1) & -x_1(1)d(1) & \ldots & x_m^\beta(1)d(1) \\ 1 & x_1(2) & \ldots & x_1^2(2) & x_1(2)x_2(2) & \ldots & x_m^\alpha(2) & -x_1(2)d(2) & \ldots & x_m^\beta(2)d(2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_1(t) & \ldots & x_1^2(t) & x_1(t)x_2(t) & \ldots & x_m^\alpha(t) & -x_1(t)d(t) & \ldots & x_m^\beta(t)d(t) \end{pmatrix}$$

and

$$[WEIGHT] = \begin{pmatrix} a_0 \\ a_{11} \\ a_{12} \\ . \\ . \\ . \\ a_{1m} \\ a_{211} \\ . \\ . \\ . \\ a_{\alpha m \ldots m} \\ -b_{11} \\ . \\ . \\ . \\ -b_{\beta m \ldots m} \end{pmatrix}$$

and

$$[TARGET] = \begin{pmatrix} d(1) \\ d(2) \\ . \\ . \\ . \\ . \\ d(t) \end{pmatrix}$$

So we have converted the learning process of the nonlinear rational function neural

network to a linear problem. It is well known that the minimum norm solution of the least squares estimation problem described in Eq.(5.14) is given by [18]

$$[\widehat{WEIGHT}] = [DATA]^{\sharp}[TARGET] \tag{5.15}$$

where $[DATA]^{\sharp}$ is the pseudo-inverse of the input data matrix, and $[TARGET]$ is the desired response vector.

In supervised learning for which the desired response $d(i)$ is provided, Eq.(5.14) can be viewed either by finding the pseudo-inverse method as shown in Eq.(5.15), or by some linear adaptive filtering algorithm [21,40] that tries to solve the linear least squares problem by developing a recursive procedure. For the purpose of building a learning machine and from a computational viewpoint, the latter procedure is to be preferred.

The crucial point to note is that the error surface is in the form of a hyper-paraboloid with a single global minimum. This is in sharp contrast to more standard neural network models in which the shape of the error surface is not known and it is not possible, therefore, to determine whether the model has converged to a local or global minimum. Another noteworthy point is that the learning phase of a rational function neural network can be accelerated by using variations of the RLS algorithm (e.g. fast recursive least squares, recursive QR-decomposition least squares) at the expense of additional computational complexity [21].

An application for which neural networks are well suited is that of classification. The main difference between function approximation discussed above and classification is that the range of classification mapping is finite. A classification function is thus non-continuous in the Euclidean sense, and so the above discussion cannot be applied directly. Since classification tries to divide the input data into several classes, it is natural to consider a classification function as a real-valued measurable function on R^m. The reason is that any function having a finite range can be expressed as

a finite linear combination of some indicator functions. An indicator function for classification purposes may be considered to be measurable. Next, we restrict the classification functions to belong to $\mathcal{L}^2(\mu)$, the square integrable function space; if this condition is relaxed, then the mean square error criterion cannot be used. Now we may present the following result :

Theorem 5.2 (Network Classification Theorem) *Rational function neural networks can approximate any classification function on compact sets of the Euclidean space.*

Proof: To prove this theorem, what we need to do is to show that continuous functions form a dense subset in the $\mathcal{L}^2$ space. Then we can conclude that rational function neural networks can approximate any classification function by the Network Approximation Theorem. More precisely, this means that for any $f \in \mathcal{L}^2$ on $[a, b]$, and any $\epsilon > 0$, there is a function g, continuous on $[a, b]$, such that

$$\|f - g\| = \left\{ \int_a^b (f - g)^2 dx \right\}^{1/2} < \epsilon \tag{5.16}$$

We shall say that f is approximated in $\mathcal{L}^2$ by a sequence $\{g_n\}$ if $\|f - g_n\| \to 0$ as $n \to \infty$.

Let A be a closed subset of $[a, b]$, and K_A its characteristic function. Put

$$t(x) = \inf |x - y| \qquad y \in A \tag{5.17}$$

and

$$g_n(x) = \frac{1}{1 + nt(x)} \qquad n = 1, 2, 3, \ldots \tag{5.18}$$

Then g_n is continuous on $[a, b]$, $g_n(x) = 1$ on A, and $g_n(x) \to 0$ on B, where $B = [a, b] - A$. Hence

$$\|g_n - K_A\| = \left\{ \int_B g_n^2 dx \right\}^{1/2} \to 0 \tag{5.19}$$

by the Lebesgue's Dominated Convergence Theorem [35]. Thus characteristic functions of closed sets can be approximated in $\mathcal{L}^2$ by continuous functions. The same is true for the characteristic function of any measurable set, and hence also for simple measurable functions.

If $f \geq 0$ and $f \in \mathcal{L}^2$, let $\{s_n\}$ be a monotonically increasing sequence of simple nonnegative measurable functions such that $s_n(x) \rightarrow f(x)$. Since $|f - s_n|^2 \leq f^2$, $\|f - s_n\| \rightarrow 0$ by the Lebesgue's Dominated Convergence Theorem again. The general case follows and the proof is completed.

Next we want to show how a neural network based on rational functions can be used to represent real-valued functions. The network has m input units and n output units. There are hidden layers that form all the polynomial combinations needed to construct the rational function of interest. Each output unit combines all the hidden and input neurons to form a rational function representation as shown in Eq.(5.3).

The input layer of a rational function neural network consists of a set of m nodes, into which we feed the components of the m-dimensional vector $(x_1, x_2, \ldots, x_m)$. The first hidden layer is designed to form all the second-order components that are common to the numerator and denominator polynomials. We should note that the desired response is also fed to the hidden layer to form the second-order components with the input vector in the training period. The second hidden layer is then assigned for the formation of the third-order components, and so on for all the other hidden layers. If, for example, a rational function with a highest order of seven in either the numerator or denominator polynomial is used, then we will have six hidden layers to get all the polynomial's combinations. For the purpose of illustration, a second-order (either α or β) rational function neural network is depicted in Fig. 5.5.

Comparing this rational function neural network to the conventional multilayer neural network, we find that there are some basic differences between their structures.

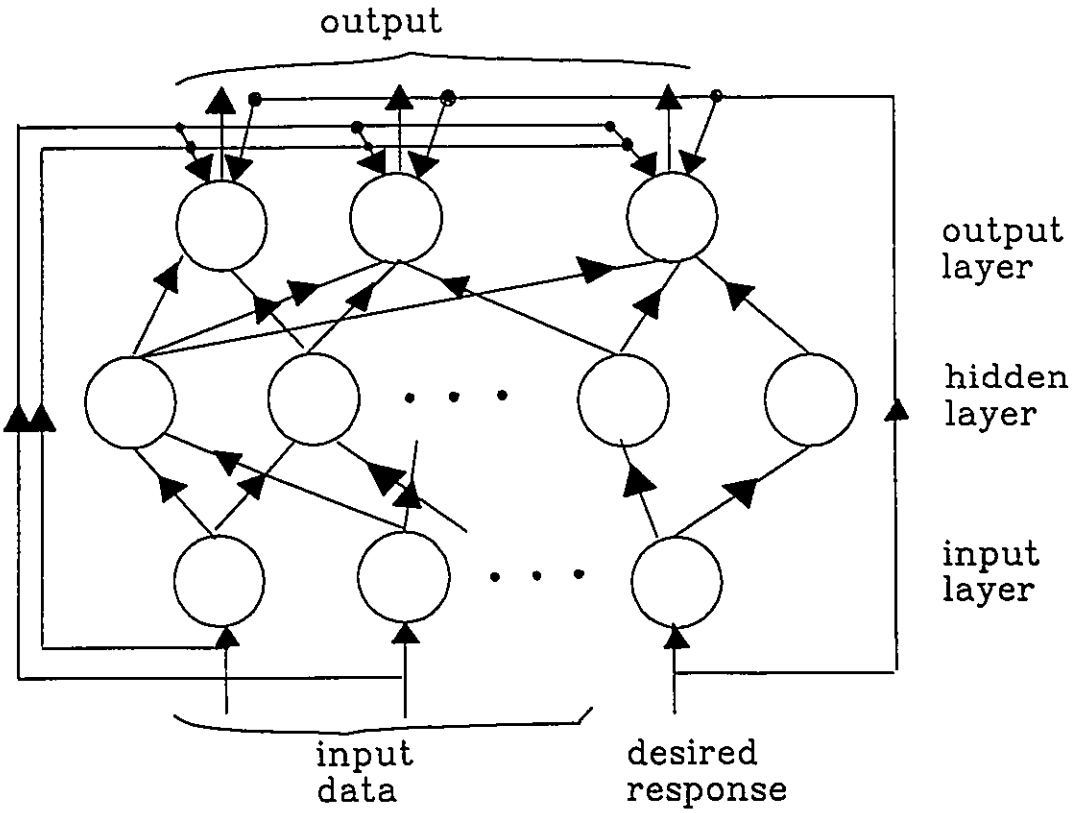


Figure 5.5: Rational function neural network

First, the rational function neural network does not backpropagate the error signal but rather the desired response. Second, adjacent layers are not fully connected in rational function neural network as they are in backpropagation and radial basis function networks. Third, the connections are not restricted to the layer-to-layer structure. For instance, the output layer is connected to all the hidden layers and the input layer. Another point of interest is that all the hidden layers are used to form the rational function, and learning takes place only in the output layer. This feature is similar to the fixed-centers radial basis function neural network, which is the reason for these two networks having a fast learning process. The main disadvantage of this approach is the complexity problem. In other words, if a rational function with a high order is used, we have to deal with too many parameters. This will cause the network to converge slowly and may affect its performance. Although the problem described herein is not as severe as that in a polynomial network, it is still the major handicap of this network. Fortunately this problem can be overcome by a method called the self-organized modeling technique [14], which optimizes the complexity of the model as well as the goodness of fit (mean squares error in this thesis). The network that we usually work with assumes a fixed structure; the self-organization technique suggests that the objective function or cost function should also include the complexity of the model. This argument makes sense, since the model order (or the number of available parameters) will appear explicitly in the measure of goodness of fit for the model. Hence the network does not only adapt the parameters but also changes the structure of the network. We will leave this self-organized rational function neural network for future study.

As for implementation, a rational function neural network is well suitable for a fully parallel and pipelined structure. Since the discovery of systolic arrays, many linear algebraic techniques have been mapped onto this fully parallel machine to achieve a fast computation. One such techniqe is the numerically stable, efficient QR-

decomposition which has also been modified by Gentleman and Kung [21] to perform the recursive least squares operation. Since the learning of a rational function neural network can be performed by recursive least squares method, this network may then be easily implemented by a systolic array (Fig. 5.6). The main modification is just a preprocessor that generates the order of basic components in a Kolmogorov-Gabor polynomial.

5.3 Application To Sea Clutter

We now apply the rational neural network to the sea clutter modeling problem. In the last two chapters, we have shown that sea clutter allows a chaotic description, and based on the dynamical theory, we know that a predictive function can be constructed to model the sea clutter. Although this sounds straightforward, one problem is that the study in chaotic analysis so far has been very ideal in the sense that the experimental data set is usually assumed to be extremely long, noise free, and simulated from some deterministic equations. Now when we are confronted with our sea clutter data, we have to accept the fact that we have no control on the data. More precisely, we can only perform the analysis purely on the data we receive from the radar and make no prior assumptions.

Our radar is a high quality instrumentation and we can believe that our clutter data set has a very high signal to noise ratio. However, noise will still be present in the data. The noise has the effect of forcing the actual low-dimensional behavior back into a high dimensional space. To compensate for this effect, we choose the embedding dimension to the the smallest integer which is larger than the correlation dimension. This is simply some compensation in the computation, and it is very hard to prove it theoretically. Since the *Takens Embedding Theorem* only gives us a sufficient condition for the embedding dimension, and hence it has no direct conflict

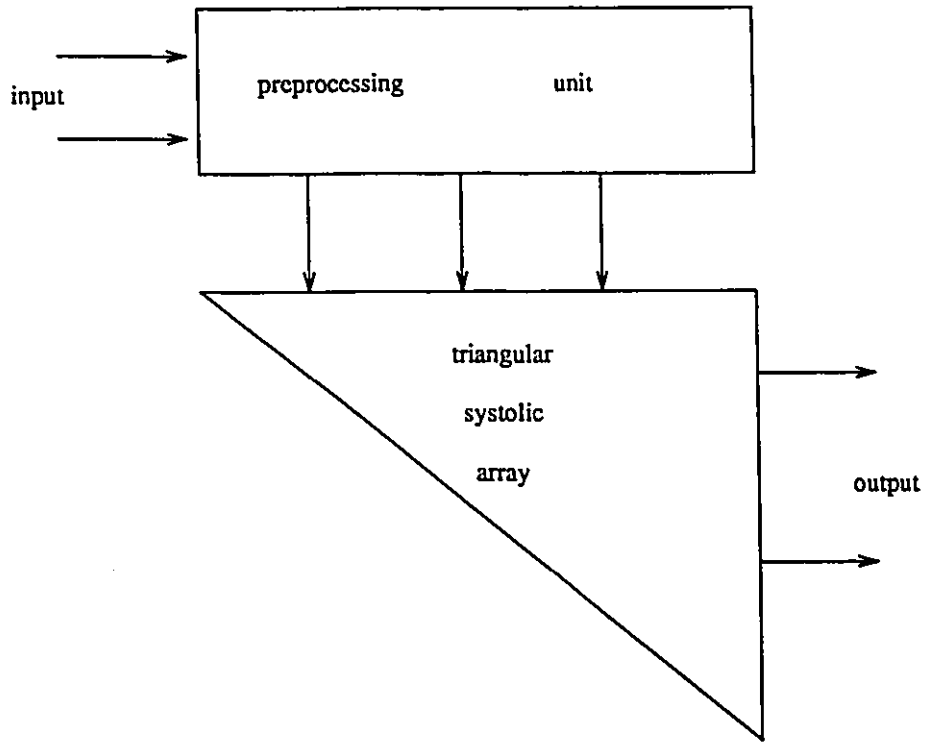


Figure 5.6: Systolic implementation of rational function neural network

with our choice of embedding dimension.

The modeling begins by using the neural network as a predictor. The way to do it is very simple. The number of input neurons depends on the embedding dimension of the process, and the output layer is usually chosen to obtain a single neuron that provides the predicted value. Note that if we want to use the network to model the dynamical system exactly, then the network should also have N input units to implement the discrete dynamical system. However, using prediction to get the next value of the process is basically the same, and can simplify the computations. Mathematically, the following equation

$$x(n) = f(x(n-1), x(n-2), \ldots, x(n-N)) \tag{5.20}$$

is mapped onto the neural network, where the function f is the underlying predictive process that the rational function neural network tries to approximate. In the learning period, $x(n-1), x(n-2), \ldots, x(n-N)$ are fed to the input layer, and the desired response $x(n)$ is fed to the input neuron as well as the output neuron as shown in Fig. 5.7. After the learning phase is completed, the network is frozen, that is, the connection weights will not be changed any more because the dynamical process is assumed to be time-invariant. This network strcture is then an approximation to Eq.(5.20), and so we may apply it to model the sea clutter data.

The rational function neural network used here has a (2,1) structure. The reason for choosing this specific structure is not totally ad hoc, although this order is not obtained by optimizing some information criteria. Obviously, (1,0) cannot be used since it is just a linear model. (1,1) is also not suitable by a recent discovery [9] that this structure cannot produce chaotic behavior since it is not sensitive to the initial condition. Thus, the simplest rational function which can generate chaotic behavior would be the (2,1) structure. Of course, a higher order structure also has the potential to produce chaos but the complexity would be greatly increased especially when the

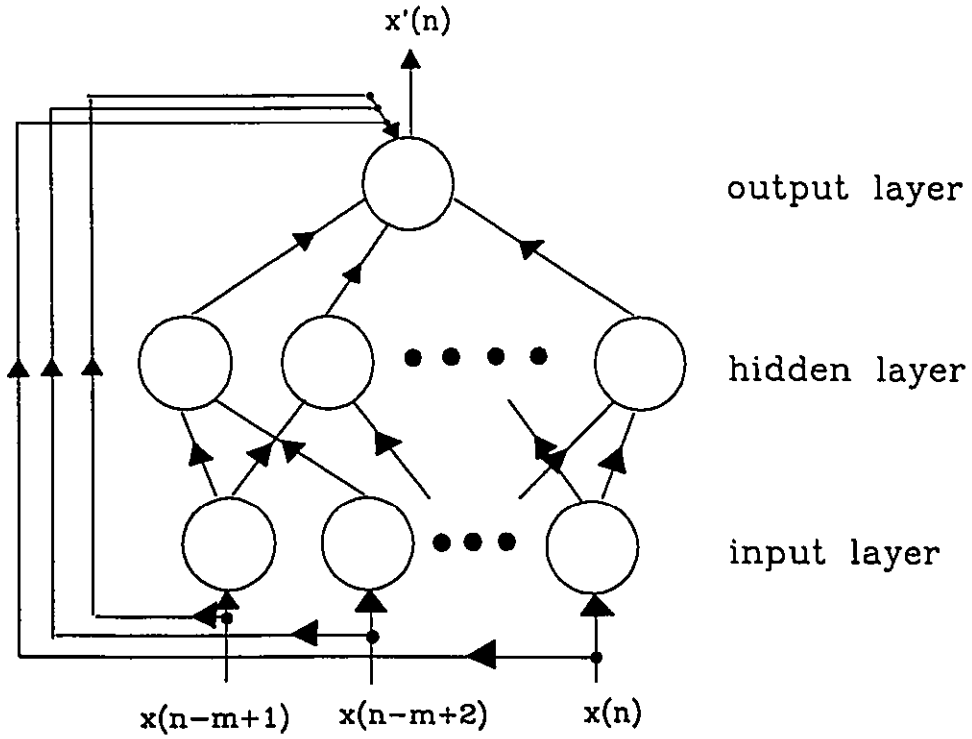


Figure 5.7: Second order rational function neural predictor

dimension of our data is resonably high (remember that the number of parameters is exponentially proportional to the order). Introducing too many parameters is not recommended by the informational Occam's razor, which suggests that the number of parameters used in a model should be minimized to keep the generality of the model provided that the performance will not be degraded.

The number of input nodes is determined by the embedding dimension found in Chapter 3. For example, the data set D1, which has a correlation dimension about 6.5, should at least have an embedding dimension seven. The second layer then contains forty nine elements which form the second-order components of the polynomials. Thus, there are a total of sixty five parameters for estimation in this structure. The learning can be done by any adaptive algorithm for linear adaptive filter as described in Section 5.1, but the LMS algorithm is chosen because of its simplicity, stability and its performance is basically the same as that obtained by a least squares technique. One thousand data points were applied to the network for training, and the result is shown in Fig. 5.8. We observe that the network does indeed converge very rapidly to an optimum level. The convergence is obtained after about 100 data points have been exposed to the network. Our computer experiment takes less than one minute on a VAX8300.

To establish that the model is good, we cannot simply just look at the training error. A small training error can only tell us that the network fits the training data which can be done in principle by any model provided that sufficient parameters are used. Hence, after the learning is completed, we have to study the ability of the network to generalize. To do this, we feed some new data to the network to observe its performance. If the prediction error is reasonably small, we can then say that the model is an appropriate one. In this thesis, we pick the "recursive prediction" to demonstrate the generalization ability of the network model. The recursive prediction uses a new input (not in the training set) to start the prediction, and then uses

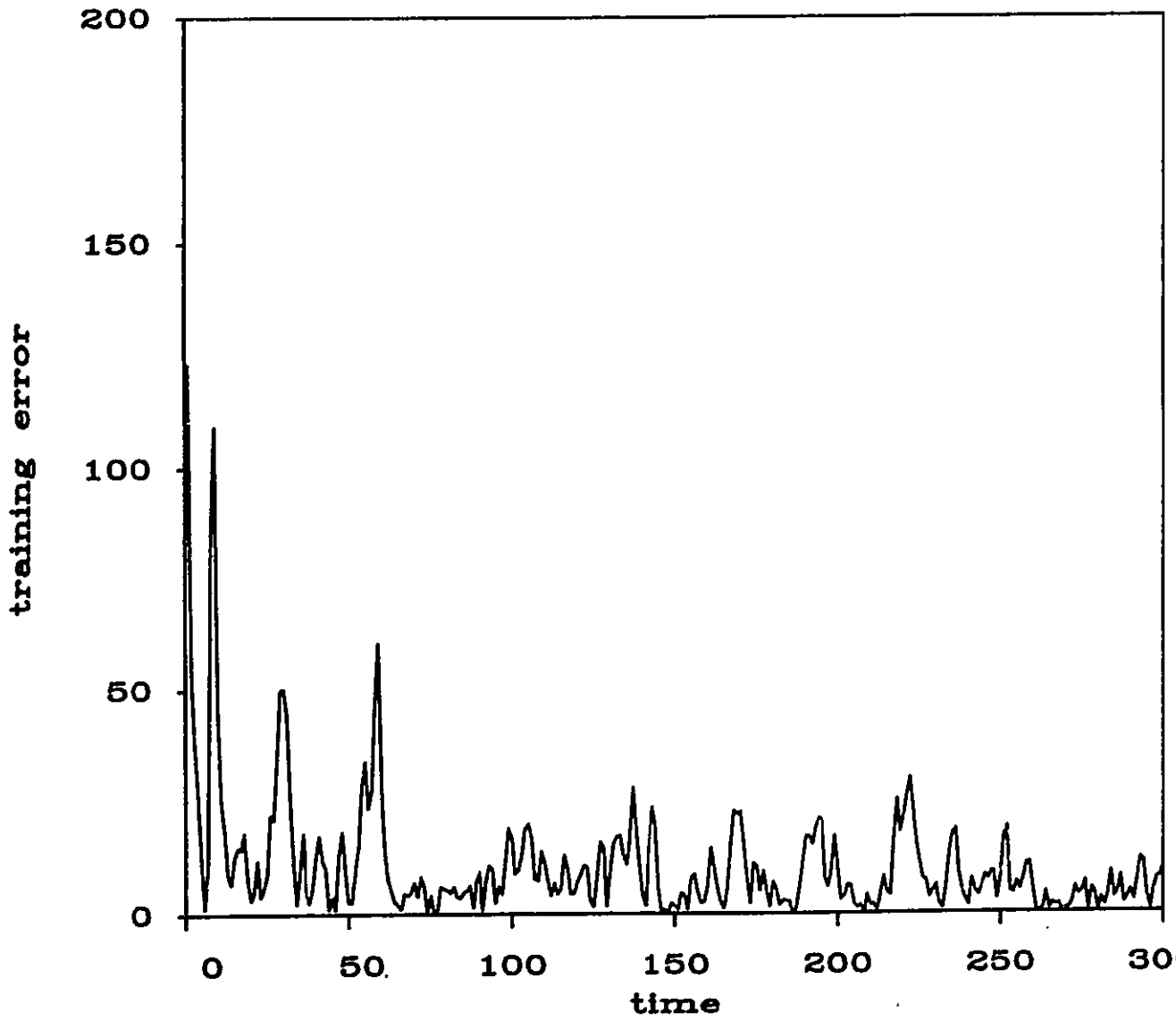


Figure 5.8: Learning curve of rational function neural predictor

the predicted point to predict the next one, and so on. In other words, the neural network predictor operates in an autonomous fashion. Ordinarily, this is a very difficult task, since there is no new information supplied to the network except for the initial starting point. The result of this operation is depicted in Fig. 5.9. We observe that the prediction errors are indeed quite small; that is, they are comparable to the training errors. We therefore conclude that this model is acceptable for the situation at hand.

The same procedure has also been applied to different neural networks for comparsion. The first one we considered is the backpropagation neural network. We may input, for example, 100 epochs each consisting of 500 data inputs. Instead, we chose to input 50,000 data points into the network so as to be consistent with the processing of data by the rational function approximation method. Unfortunately, this attempt did not seem to be successful. The recursive prediction behaves poorly. Sometimes the network produces a straight line (in dynamics theory, this means that the system has a stable fixed point as the attractor, which is impossible for our data), and sometimes it produces some behavior that is totally unrelated to the actual process.

The second comparison is based on the radial basis function neural network. Among the three versions of this network, the nonlinear structure is similar to the backpropagation network; hence it is ignored. For the purpose of comparison, we chose the hybrid network, since it usually has a better performance than the fixed-centers network. The result is plotted in Fig. 5.9 to provide comparison with the performance of the rational function neural network. We observe that the performance of this approach is about the same as the rational function neural network. However, the complexity is much higher in the hybrid radial function network than in the rational function network. Not only does the radial basis function network need to compute the complicated Euclidean distance of high dimensional vector, and

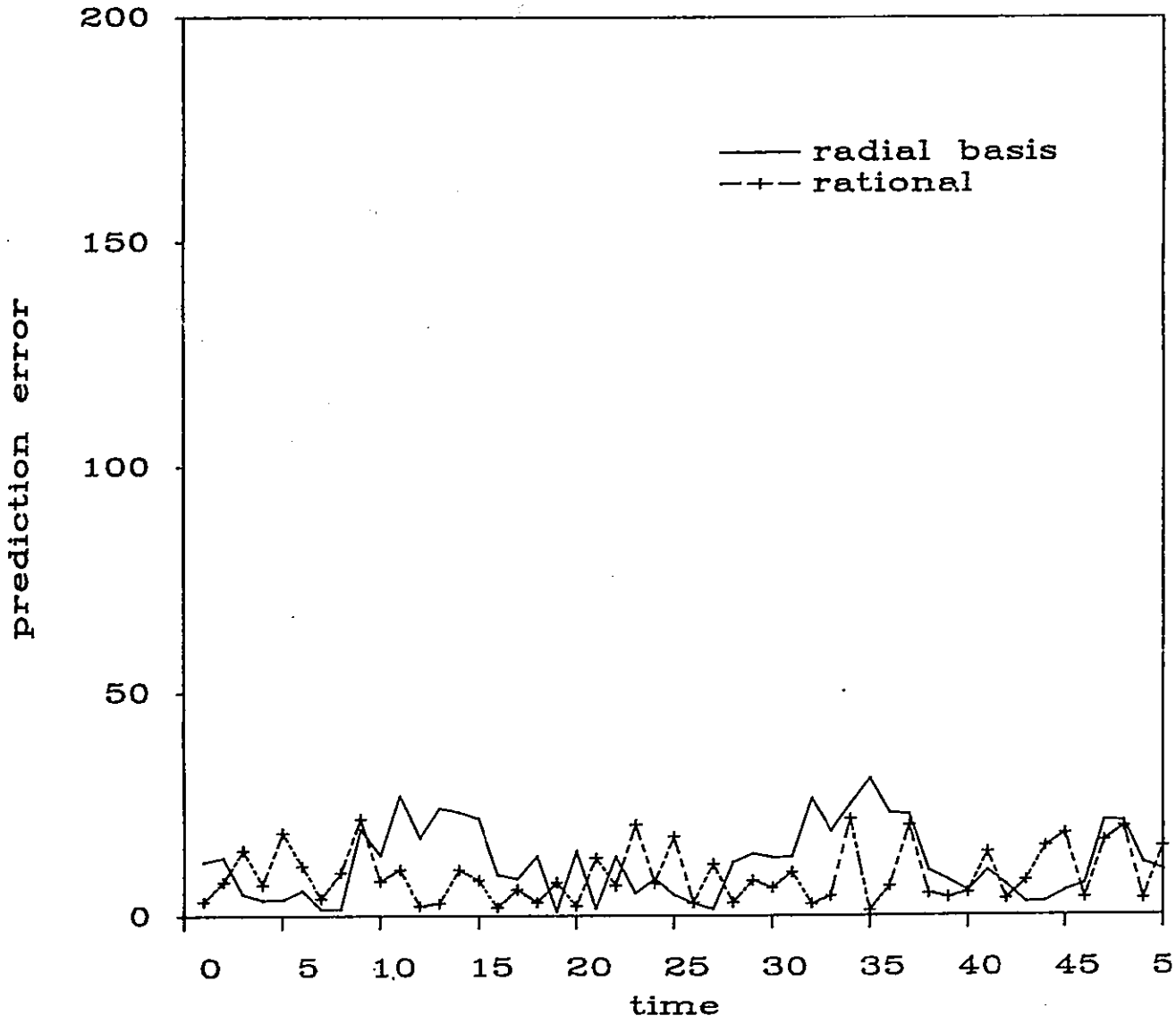


Figure 5.9: Prediction error of sea clutter
mean value of D1 $\sim$ 130

a time consuming k-means algorithm is involved, but the hybrid structure also uses 300 to 400 hidden units to obtain the performance plotted in Fig. 5.9. On the other hand, the rational function network only needs sixty five parameters. Therefore, the learning rate of the hybrid radial basis function network is much slower than the rational function network.

The last comparison is based on the polynomials method. Here we chose a second-order Volterra structure in light of the complexity of the problem at hand. (With the dimension of the input vector being seven, a third order structure needs 343 more unknown parameters.) We find that the performance is not very consistent. More precisely, the recursive prediction sometimes works (though not as good as the other two), but sometimes the recursive prediction can make the the network diverge as shown in Table 5.1. This phenomenon may be a result of the poor extrapolation property discussed in Section 5.1.

In this chapter, we try to build a model for sea clutter. The way to reconstruct the dynamics is to employ a neural network as a predictor. After the network predictor is trained, the dynamics of sea clutter is described by the recursive prediction. There are many neural network structures existing in the literature, and the most popular and powerful one is the backpropagation network. However, this network in its present form is not suitable for real-time application. There is definitely a need for an efficient neural network structure for the use in signal processing. In regard of this need, we construct a novel neural network structure called rational function neural network, which has the same computational power as the backpropagation network, and it can be trained linearly (that is, there is only one local minimum). We apply this network to our sea clutter, and the experiment results tell us that rational function neural network forms an excellent reconstruction of sea clutter dynamics, and the training of this network is just as effficent as the training in any linear adaptive filter.

i	prediction error
1	19.50423
2	20.71204
3	7.791940
4	0.731318
5	384.0915
6	36.48622
7	271.5546
8	0.619229
9	303.0548
10	414.9706
11	528.2668
12	868.3160
13	1295.218
14	1927.697
15	5877.921
16	5894.656
17	14689.86
18	15952.43
19	51228.14
20	56549.02
21	225036.3
22	354819.8
23	1352786.
24	709412.4
25	1.4326E+08
26	2.1683E+11
27	4.0574E+17
28	1.5038E+30

Table 5.1: Prediction error sea clutter using second order adaptive Volterra filter

Chapter 6

Memory Based Neural Network

6.1 Memory Based Reasoning

The prediction discussed in the last chapter relies on a fundamental concept, that is, the assumption of the existence of an underlying rule which can be approximated by some functions. A neural network attempts to extract this underlying rule from the experimental data by its functional structure. This approach has its root in human beings' prediction. We call this approach prediction by learning. Human beings increase their knowledge by learning the underlying concept and hence try to predict the problems by this learned concept. However, when we think about ourselves more carefully, we know that we also have another ability to perform prediction and it is based on the ability of memorization.

Most of the research done in neural network theory is concentrated on the learning approach. The backpropagation network, perceptron, radial basis function network and the rational function neural network developed in Chapter 5, are examples of neural networks based or this learning idea. More precisely, these networks assume some kind of functional form and try to make it a universal underlying rule to retrieve

or approximate the exact cause of a particular problem. We do not reject the utility of this approach. But, we believe that the theory using memory warrants further studies. First, it is difficult to conceive of thought without memory. Second, a human being can perform many tasks by the use of so-called common sense which is based essentially on undigested memories of past experience.

The basic idea of using memory to construct a neural network can be roughly described as follows. First, we count the number of times various features or combinations of features occur. For example, in the context of a medical application, if we have a patient with a high fever, we would want to find out how often the high fever occurred in combination with various diseases. Mathematically, this step constructs a relationship between the target or desired response with the given inputs. Second, we use these feature counts to produce a metric to compare the distance between two features. The reason for this is to find the closest feature in the past experience to match the new feature which is the fundamental idea of memory approach. So the third step is to calculate the dissimilarity between each item in memory and the current case, and the last step is retrieving the best matches. In the real world, there may be no exact matches, so the best match is called for.

Although the basic idea of memory based approach is simple, its implementation is somewhat complex. First, the storage of a large amount of data is quite expensive. Second, there is no general way to search memory for the best match without examining every element in memory. On a Von Neumann machine, this makes the use of large memories impractical. Fortunately, with the development of new parallel architectures, specifically the connection machine system, the operations necessary to implement this appraoch have become sufficiently fast to allow experimentation.

To illustrate the memory based idea, consider the task of solving simple algebraic equations in one variable. Each instance of this task is an equation involving only one variable and the four arithmetic operators. A solution is an equation of the form

$y = E$, where E is an expression containing only constants. For example, given the problem $6 = 4 \times y$, the solution is $y = 6/4$. A simple memory system would memorize the pair $(6 = 4 \times y, y = 6/4)$. We can also make the system to memorize the generalized pair $(A = B \times y, y = A/B)$. To this pair, we must attach three applicability conditions : A and B must be constants and B must not be equal to zero.

When a new problem, $3 = 2 \times y$ is presented to the system, it matches the stored pattern (with substitution $\{A/3, B/2\}$). Furthermore, the three applicability conditions are satisfied. Therefore, the solution can be constructed by retrieving the stored solution pattern to obtain $y = 3/2$. If there is no memorized pair that matches the new problem, then the network would look for one in the stored pattern which is very similar to the new problem.

From the four basic steps described above, we can see that our memory based neural network is basically accomplished by searching a database of worked problems for the the best match to the problem at hand. This requires a means of judging how closely two situations match, leading to the topic of metrics. A metric is a measure of distance and retains the same meaning used in mathematical analysis. The implementation of memory based neural network depends on finding a suitable definition of the metric function.

Before we can discuss the metric, we need to define a record as a structured object containing a fixed set of features. The simplest metric would then be the number of features for which two records have different values. Although this metric makes a convenient starting point, it is not very useful because it assigns equal weights to all features. A better metric would incorporate the importance of features by giving different weights to different features.

We will not go any further on the discussion of measuring similarity between two general objects since our attention is given to systems that learn using entries other

than symbolic ones. More precisely, our entries, sea clutter data, are formatted as vectors to be reconstructed as points on a manifold embedded in Euclidean space. From now on, we will develop the memory based neural network in the Euclidean space. We mention the importance and construction of metric for general objects simply by highlighting the fact that this approach is not restricted to numerical data only.

When we apply the memory approach to our sea clutter problem, the following geometrical picture should be kept in mind. By phase space reconstruction, the data are supposed to form a manifold in the Euclidean space. The training data are then the database and form the memory. When we get a new entry, we then look for the "closest" one in the memory to the new entry, that is, two points very close in the Euclidean space. Hence, the metric function can be any metric in Euclidean space such as absolute norm, Euclidean norm, or Minkowski norm. The one used here is the Euclidean metric since it is smooth and most popular. The use of other norms should have some advantages over the Euclidean metric in certain applications and it is left for future research.

The basic idea of adaptive filtering is to obtain the relation between the input and the desired response. Thus, the target value of a new entry should be strictly related to the desired response of the corresponding closest element in the memory database. A simple choice would be direct substitution. That is, the desired response of the closest element is taken to be the target value of that new entry. This choice seems valid because these two points are supposed to have similar behavior and that is why a good metric function is needed.

In our case, sea clutter data are known to lie on an attractor or an invariant manifold. The above approximation is simply an approximation which tries to approximate the underlying rule locally. In other words, we believe that two points which lie very close on a manifold have similar behavior. Putting it in mathematical

language, if we have $|a-b| < \delta$ where δ is a very small number, then the target values of a and b given by c and d respectively should also be very close, $|c-d| < \epsilon$ where ϵ is a very small number. Assuming some functional relationship between the target and input, we have $c = f(a)$ and $d = f(b)$ and the above closeness argument is similar to the continuity definition of a function. We can now accept that this memory based approach applied to our problem would produce a reasonable approximation to the underlying rule.

There are some comments we would like to make at this stage. First, the assumption of the continuity of the function f is a very mild one in adaptive filtering and neural network applications. The classical adaptive filter, that is, the linear model, and the most popular multilayer perceptron inherently assume that f is infinitely differentiable. Second, in a dynamic reconstruction problem or prediction problem, if we know that an attractor exists in the data, the above closeness argument even makes more sense. The reason is that an attractor is attracting the evolution of a dynamical system, the target values of two close points would get even closer as illustrated in Fig. 6.1. Our sea clutter data admit an attractor even though it is a strange one. We can still believe that the next evolution of two close points should be quite close too. Third, because of the existence of noise in our data, taking simply one closest point may be very dangerous. We therefore suggest the use of several points that are very close to the new entry to compensate the noise effect. In other words, we attempt to construct a neighborhood of the new entry rather than a single point. Fourth, with a set of points in the memory database, we can then perform approximation by some better technique such as linear regression.

Roughly speaking, the memory approach does not attempt to build a closed functional form to fit the data but rather approximate the underlying rule by having a good local fit. The basic idea is to break up the domain of the underlying rule, which is known to be a nonlinear function f in our chaotic problem, into local neighbor-

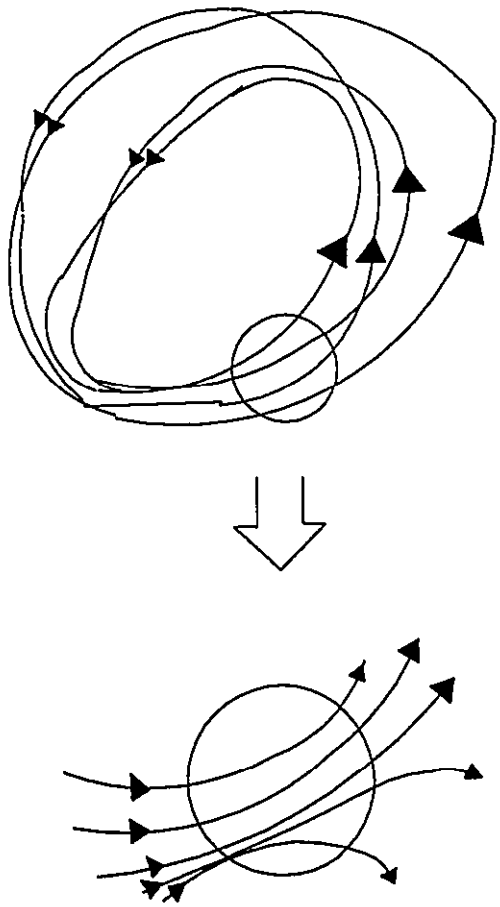


Figure 6.1: Evolution of nearby trajectories around an attractor

hoods and fit different parameters in each neighborhood. Fitting different parameters in each neighborhood means that we can perform different fitting techniques in a local neighborhood such as averaging, linear fit and quadratic fit. When f is smooth the neighborhoods can be small enough so that f does not vary sharply in any of them, making the constraints of a particular representation less important.

Comparing the memory based neural network to the learning based neural network, we have the following observations. First, the memory approach is more general than the learning approach in the sense that fewer statistical and geometrical assumptions about the data are required. Consequently, the memory method can be successfully applied to a wider class of behavior. Second, to increase the accuracy in the learning method, we require a higher order approximation, and it often strongly increases the complexity. However, statistical estimation theory and computational experience tell us that the performance would quite often be degraded. In the memory approach, adding more local neighborhoods is used instead of adding higher order, and adding more neighborhoods is obviouly more robust and efficient. Third, the memory approach usually produces better fits for large data sets, and the learning method would have a better performance when the size of the training set is small. Fourth, the memory approach does not provide a closed form expression and therefore is not as convenient as the learning method for performing further analysis. Fifth, memory approach is more trouble to implement than global approximation.

6.2 Prediction on Manifold

Memory based approach makes it possible to use a given functional representation efficiently and hence it can be used to predict the sea clutter evolution. The four general steps listed in the previous section can be summarized into the following procedure for the application to our problem. First, we need to assign neighborhoods

and next, we look for a mapping that maps the points in each neighborhood into their future values. To make a prediction we evaluate the chart at the new entry.

There are many ways to assign neighborhoods. One way is to partition the domain of f into disjoint sets. So when there is a new entry presented to the network, the network will locate the new entry to a suitable region and hence the neighborhood is defined. The advantage of this assignment is a fast searching procedure. Comparing distances is the most time consuming process in the memory approach; this partition can efficiently locate a new data point according to its coordinates and hence those training vectors in this region can be retreived from the database quickly. The disadvantage is that this partition may give poor performance, since there is no overlap between the neighborhoods, and therefore no continuity between charts. A point near the boundary of its neighborhood will be poorly approximated.

Although the above assignment is convenient, it does not match the memory idea very well. The memory approach requires the search for those points in the database that are closest to the new data point but not just some points quite close to it. Also the approximation ability using a fixed partition is reduced. Therefore, we would construct the neighborhood for a new point by choosing data points in the database which are closest to the new entry. Let $\{\vec{a}_i\}$ be the training vectors. We want to find a neighborhood for a new input vector $\vec{x}$. Assuming that the number of elements in the neighborhood is m, the criterion is to search for m vectors belonging to $\{\vec{a}_i\}$ which minimize $\sum \|\vec{x} - \vec{a}_i\|$.

One remark is that although this criterion seems to provide a set of "closest" data points in the training database, does it mean that the prediction based on this neighborhood is optimal? We suspect that the answer is negative. Consider a point in a two dimensional plane; if a three-point neighborhood is chosen and the point does not lie inside the triangle formed by those three points, the approximation may be poor. This tells us that a neighborhood formed purely from a closeness criterion

may not give us the best performance. To overcome this problem, we may try other metrics or put constraints on the cost function. However, we will use the closeness criterion in this work only, since in general it will give us a reasonable result. Adding constraints will greatly increase the complexity of the problem. This situation is similar to that in linear regression which has a poor performance when the predicted value does not lie in the region spanned by those regression points, and this problem is usually neglected in general applications.

The second problem is to find an efficient way to locate all of the points closest to a given reference point. The dynamical embedding method imposes stringent requirements on any nearest neighbor algorithm. The storage overhead for the corresponding data structures must be small, because there are tens of thousands of attractor points. The algorithm must be fast, since there is one nearest neighbor problem for each new entry.

Finding the k nearest neighbors is a time-consuming procedure, particularly for a large number of training samples. Hence there are several strategies that have been proposed to reduce the complexity of this process. One way to do it is to introduce a preprocessing step to reorder the training samples so that each sample tends to be far from its predecessors in the ordered list. The nearest neighbor of a test sample is found by the training samples in the listed order. The list can also be ordered according to their projection values on a given coordinate axis. Then the search proceeds by examining the training samples in the order of their projection distances from the test sample until the projection distances becomes larger than the full N-dimensional distance to the kth nearest neighbors among the training samples already examined. Other well-known techniques such as the ordered partition technique, which partitions the training samples by their coordinate values along each axis and uses the branch-and-bound search method, resulting also in a considerable reduction of the number of examined samples. All these techniques attempt to reduce the number of distance

calculations and show considerable reduction in the computation effort.

The neighborhood finding technique used in this thesis is the well known $k-d$ tree method [5]. A $k-d$ tree corresponds to a decomposition of a k-dimensional Euclidean space into hyper-rectangles. It is an ordinary binary tree, but each internal node has room for both a discrimination value and a dimension number. The root of the tree represents the entire space. The dimension number assoicated with the root specifies the first dimension along which the space is cleaved. The discrimination value specifies the location of the cut along this dimension. If the root's dimension number is i and its dimension value is v, then the decomposition of the space is by the hyperplane defined by $x_i = v$. Its left child represents the half space of all points satisfying $x_i \leq v$, and its right child the half space of all points satisfying $x_i > v$. In general, each node of a $k-d$ tree corresponds to a hyper-rectangular piece of the space in which some of the dimensions may be infinite. The set of nodes at a given level of the tree correspond to a set of hyper-rectangles which partition the space. The leaves of the tree correspond to hyper-rectangles forming the finest partition of the space.

Searching a $k-d$ tree for the leaf bucket containing a query point is easily accomplished in a single traversal of the tree from root to leaf. At each node, we compare the value of the point's coordinate in the dimension specified in the node with the discrimination value stored in the node. The traversal proceeds to the left or to the right depending on the outcome of this comparison. A well-built tree will have only $log_2 n$ levels when there are n hyper-rectangles in the leaf partition. The search time is then logarithmic in the number of stored entries.

If we want to find the nearest neighbors this make it possible to eliminate many points from consideration without actually computing their distances. The $k-d$ tree has the nice property that it flexibly partitions only the parts of the space that actually contain data, adding partitions only where they are needed. This is a crucial point that makes this memory based method suitable for real-time signal processing

applications.

Once we have chosen neighborhoods, the next step is to find a way to perform prediction. There are many ways to achieve this goal. In those chosen neighborhoods, the future values of all the neighbors are known because all the data points in the given time series have a point coming after except the last point, and hence the exact future values of all the training vectors are known. We can also think of this as a supervised training scheme, where the desired response is given. Since all points in a neighborhood are very close and by the memory idea or the attracting property of dynamical theory, we can expect the future value of the new entry to be strongly related to the future values of those neighbors. By the intuitive memory idea, the future value of the closest training point is taken to be the prediction of the new entry. Now, we have a set of close points and hence the prediction can be taken as the average of the future value of the training points in a neighbor.

Remember that our embedded dynamical system is a manifold in Euclidean space. A neighborhood is a very small region on the manifold and hence can be considered to be a subspace of curvature zero. In other words, it is just a normal Euclidean space where the dimension is lower than the one in which the manifold embeds. Fitting a local chart becomes a very standard statistical regression problem. Therefore, the future value of the new entry can be taken as the average of the future values of the training vectors in the neighbor. This approach can give us a very robust and fast prediction of a new entry. There are many other techniques which can be used to perform the regression. For instance, the most popular is the linear regression technique which usually has an acceptable performance. The quadratic fit is another alternative, which is also reasonable. Other more sophisticated data fitting methods such as the spline technique, or the radial basis function can also fit the scattered data accurately.

The elegance of proving that sea clutter data is chaotic is in the smooth mapping

found to generate the data. The estimation can then be improved by increasing the order of approximation, at least in principle. For instance, a quadratic polynomial can approximate an unknown function better than a linear polynomial. However, there is a tradeoff in the sense that higher order approximations need more data points. In other words, a larger neighbor is required to provide sufficient data for high order approximations. The theory that the higher the order, the better the performance is true in mathematical theory only; that is, assuming that an infinite number of points are provided. In practice, the number of data points is an important decisive factor in choosing a suitable fitting method. In our situation, the correlation dimensions are quite high and hence the embedding dimensions would be high also. Therefore, we need many points to have a very "nice" neighborhood. That is, we need many such neighborhoods to fill up the embedded manifold. Because of the nonstationarity property of sea clutter and our limited data sets, we should not use a very long data sequence and hence only averaging and linear regression are used in this thesis. In fact, since the model is built on real data which cannot be noise-free, it would be wise to decrease the accuracy so as to retain robustness [7].

To summarize the procedure, first we find a neigborhood and then we construct a map for the data in the neighborhood to perform prediction. For those who are familiar with topology, they will find that this procedure is closely allied to differential topology. In signal processing applications, the most widely used mathematical techniques are statistics and algebra. The possiblity of using topology has never been exploited. Actually this is a result of the common assumption that processes in signal processing are random. To see the relationship between our method and differential topology, we will briefly explain some basic topology ideas which are strictly related to the memory method. A set M is given the structure of a differentiable manifold if M is provided with a finite or countable collection of charts, so that every point is represented in at least one chart. A chart is an open set U in the Euclidean coordi-

nate space $\vec{q} = (q_1, q_2, \ldots, q_N)$, together with a one-to-one mapping φ of U onto some subset of M, $\varphi : U \to \varphi U \subset M$. We assume that if points $\vec{p}$ and $\vec{p'}$ in two charts U and U' have the same image in M, then $\vec{p}$ and $\vec{p'}$ have heighborhoods $V \subset U$ and $V' \subset U'$ with the same image in M. In this way we get a mapping $\varphi'^{-1}\varphi : V \to V'$.

This is a mapping of the region V of the Euclidean space $\vec{q}$ onto the the region V' of the Euclidean space $\vec{q'}$, and it is given by N functions of N variables. The charts U and U' are called compatible if these functions are differentiable. An atlas is a union of compatible charts. Two altases are equivalent if their union is also an atlas. A differentiable manifold is a class of equivalent atlases. A neighborhood of a point on a manifold is the image under a mapping $\varphi : U \to M$ of a neighborhood of the representation of this point in a chart U.

Now we see that the conventional characterization of a manifold by neighborhoods and charts is surprisingly equivalent to the idea of using memorization. In other words, the memory method tries to construct a local characterization of the manifold containing the data. Note that this is not always the case since the existence of the manifold is based on the *Takens Embedding Theorem* which applies to data with finite dimension only. For a purely stochastic process, the memory idea may still be used but this geometrical figure may not be a valid indication of the real picture (we may need to use more complicated geometric theories such as the stochastic manifold and stochastic differential geometry).

In this thesis what we are attempting to do to the radar data is to perform dynamic reconstruction. So, what are we exactly doing with this memory method from the point of view of differential topology? To answer this question, we need to know what a dynamical system is. One of the most important ideas in differential geometry is that of a tangent space to a manifold which is based on the intuitive geometric idea of a tangent plane to a surface. We define that two curves are tangent to each other at a point p in M if they both cross the point p and in some local coordinate system

around the point the two curves are tangent in the usual sense as curves in R^N. A tangent vector at p is an equivalence class of curves in M where the equivalence relation is that two curves shall be tangent at the point p. The tangent space T_pM to M at the point p is the set of all tangent vectors at the point p. The union of all these tangent spaces $TM = \bigcup_{p \in M} T_pM$ is called the tangent bundle, which is a very special case of the general idea of a fibre bundle. The importance of a tangent is that a tangent vector can be used as a directional derivative of a function on M, which enables the equivalence class of curves to act as a type of differential operator on the space $C^\infty(M)$ of real-valued differentiable functions on M. We then think of $C^\infty(M)$ as a ring over R, and of a tangent vector as a derivation map from this ring into R. Therefore, a dynamical system is described by the tangent vector or more generally the tangent bundle.

What we do is look locally at the invariant manifold M and attempt to follow the dynamics in various regions of the manifold by projecting the orbit into the tangent space T_pM of M at various points p of M. The basis in each tangent space will be chosen according to some algorithms and will reflect the particular dynamics taking place on that part of the attractor. The transition matrices, which describe how the coordinate systems are related in regions of overlap of the charts, will tell us how the nature of the dynamics is changing. In part of the attractor where a particular set of spatial structures dominate, the tangent space will be spanned by the optimal basis which will be the generators of the infinitesimal motions of the modes characteristizing the mutual and self interactions of these dominant structures. As the orbit traverses the invariant manifold, the chosen bases will undergo a sudden change from one set of structures to another. The transition matrices identify the change in the dynamics. When the change is dramatic, the transition matrices diverge maximally from the identity. They serve therefore to help us understand both the geometry of the attractor and the physical nature of, for example, the rare heteroclinic

like transitions between different regions of the attractor. To apply this topological structural idea to experimental data, we require a hugh number of data to perform the transition analysis. The reason is that the transition can be described by matrix theory only when we look at the manifold in a very flat region. If the data are not cloud enough, the transition on a manifold requires the use of an affine connection and the search for geodesic, which will greatly complicate this topological study. Due to the practical problems and nonstationarity of sea clutter, we will not study this interesting transtion problem in this thesis.

Now the topological picture is quite clear. The experimental data can be reconstructed to form a manifold, and the memory method can be considerd to be a procedure to estimate the tangent bundle. Because of the quality of the experimental data, constucting a good local coordinate of all points on a manifold seems impractical (in other words, it is very difficult to get the dynamics completely). However, the prediction based on this topological method usually has an acceptable performance.

The proof that the memory method applied to the dynamics reconstruction problem always works in principle for smooth dynamical system relies on the Morse Lemma. Suppose that f represents a smooth real-valued function on M, that is, $f : M \to R$. At a particular point $\mathbf{x} \in M$, f is either regular or $df_{\mathbf{x}} = 0$. If it is regular, then we can choose a coordinate system around $\mathbf{x}$ so that f is simply the first coordinate function. Thus we really know all about the local behavior of f at regular points, at least up to diffeomorphism. The problem then comes from those critical points. By Sard's Theorem, we know that the set of critical values of a smooth map on a manifold has measure zero. However, critical points always exist for every function. In our discussion we will only consider nondegenerate critical points since any bounded smooth function can be uniformly approximated by a smooth function, which has only nondegenerate critical points. The Morse inequalities place restrictions on the number of critical points that the function f can have due to the

topology of M, and the local behavior of a function at a nondegenrate critical point is completely determined, up to diffeomorphism, by the Morse Lemma.

Lemma 6.1 (Morse) *Suppose that the point* $\mathrm{a} \in R^N$ *is a nondegenrate critical point of the function* f, *and*

$$(h_{ij}) = \left(\frac{\partial^2 f}{\partial \mathrm{x}_i \partial \mathrm{x}_j}(\mathrm{a}) \right) \tag{6.1}$$

is the Hessian of f *at* a. *Then there exists a local coordinate system* $(\mathrm{x}_1, \ldots, \mathrm{x}_N)$ *around* a *such that*

$$f = f(\mathrm{a}) + \sum h_{ij} \mathrm{x}_i \mathrm{x}_j \tag{6.2}$$

near a.

Thus every function near a nondegenerate critical point is locally equivalent to a quadratic polynomial, the coefficients of which constitute the Hessian.

The Morse Lemma can be applied to any smooth map f and hence this topological idea is not only restricted to prediction. The function f can be chosen to perform any filtering operation. Conventionally, filtering, no matter it is linear or nonlinear, is modeled by a stochastic equation, since some stochastic factor must be used to model the random behavior of natural physical process. However, chaotic theory tells us that dynamical system itself can generate highly random behavior without introducing any pure stochastic noise. In view of this, filtering can now be formulated into some deterministic dynamical system, and those conventional stochastic differential/difference equations can be replaced by some differential/difference equations. In particular, adaptive filtering, in conventional language, is a technique which tries to find a functional relation between the input random variables and the desired response (another random variable). Taking the unnecessary statistical assumption out, adaptive filtering is just an adaptive technique which tries to find a mapping between input data and desired response; that is, a function approximation problem. By the Morse Lemma, if the data input can be embedded on a manifold, the filtering

function can be estimated accurately (a quadratic polynomial is sufficient; of course, other more advanced approximation technique such as radial basis function may be better in practical sense). We call this filtering method as **adaptive filtering on manifold**. The key issue is the possibility of finding an embedded manifold. This can be guaranteed by using *Takens Embedding Theorem* when the data exhibits deterministic behavior. This new approach for adaptive filtering is called **chaotic adaptive filtering** to highlight the underlying deterministic assumption.

6.3 Prediction on Simplices

The chaotic adaptive filtering discussed above assumes that the received data form a manifold. Although the existence of a manifold is true in principle, it may not be exactly the case in practice. The reason is that the data set must be a finite set, and the manifold and topology theory assumes continuity. What we really have is a sequence of vectors which is often treated as merely a cloud of points in N-dimensional space. So we are confronted with an immediate question : Can the theory be applied to a finite set of vectors rather than a manifold? We may also consider this finite set of vectors as an approximation to the actual manifold, and try to understand the validity of this estimation. As this stage we would like to introduce the simplicial approximation to a dynamical system.

Consider the problem of explaining the difference between a sphere and a torus. The difference, of course, is apparent : the sphere has one hole, and the torus has two. From the homology viewpoint, every simple closed curve on the sphere is the boundary of the portion of the spherical surface that it encloses and also the boundary of the complementary region. However, a meridan or parallel circle does not separate the torus. Thus any simple closed curve on the sphere is homologous to zero, but meridian and parallel circles on the torus are not homologous to zero.

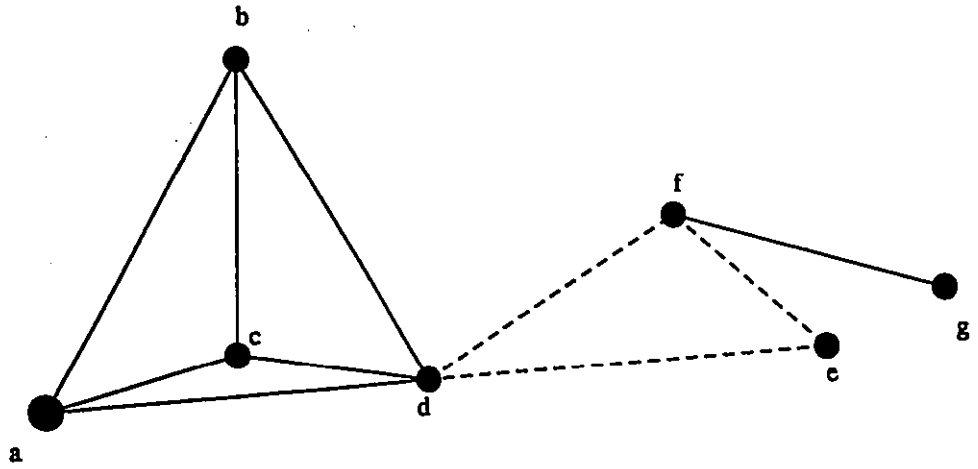


Figure 6.2: A polyhedron

The following intuitive example will make more precise this still vague idea of homology. Consider the configuration shown in Fig. 6.2, consisting of triangles $< abc >$, $< bcd >$, $< abd >$, and $< acd >$, edges $< ab >$, $< ac >$, ... $< ef >$, and $< fg >$, and vertices $< a >$, $< b >$, $< c >$,... and $< g >$. The interior of the tetrahedron and the interior of triangle $< def >$ are not included. This type of space is called a polyhedron. Polyhedron does not only provide an approximation to a surface, but it is also the most fundamental and common technique in algeraic topology. So far we have only mentioned the relationship between differential topology and dynamics; in fact, differential topology connects the dynamics and the chain complex description, and algebra can relate the chain complex description with homology. In particular, if the dimension is greater than five, we should convert the dynamical study into the algebraic topology of the manifold. Our sea clutter attractors have been shown to have a dimension higher than five, algebraic topology would therefore be a very poweful tool in studying the sea clutter manifold. We will leave this for a future study.

A polyhedron can be regarded as a subspace of some Euclidean R^N which is obtained by properly gluing together certain elementary spaces called simplexes. A m-simplex (written σ^m) is a generalized triangle in m-dimensions. Thus σ^2 is a triangle and σ^3 a tetrahedron. The simplexes are glued together in such a way that two simplexes, if they meet, have a common vertex or edge. More precisely, an m-simplex, σ^m , is the set of points $\mathbf{x}$ in R^N given by :

$$\sigma^m = \left\{ \mathbf{x} = \sum_{i=1}^{m+1} \lambda_i \mathbf{x}_i \mid \lambda_i \geq 0, \sum_{i=1}^{m+1} \lambda_i = 1 \right\} \tag{6.3}$$

where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m+1}$ are distinct points in R^N which are independent. We often write $\sigma^m = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m+1}]$ and call $\mathbf{x}_1, \ldots, \mathbf{x}_{m+1}$ the vertices of the m-dimensional simplex (or m-simplex) σ^m. The λ_i are called the barycentric coordinates of the simplex.

We note that the point $\mathbf{x}$ contained in m-simplex, σ^m, corresponding to a given set of barycentric coordinates $\lambda_1, \lambda_2, \ldots, \lambda_{m+1}$ can be regarded as the centre of mass of the system with masses $\lambda_1, \lambda_2, \ldots, \lambda_{m+1}$ placed at the vertices $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m+1}$ respectively. This physical analogy leads us to expect that if all the λ_i are non-zero, then the corresponding set of points $\mathbf{x}$ represent the interior of σ^m, while if any $\lambda_i = 0$ then the set of points $\mathbf{x}$ represents a face of σ^m opposite to the vertex $\mathbf{x}_i$

Assume that we are given an N-dimensional manifold, and that the manifold is mapped to itself by the dynamical system. In constructing a simplicial map, the manifold is first divided into a chosen grid of N-dimensional hypercubes. Each hypercube is in turn conceptually divided into $N!$ simplices. A simplex in an N-dimensional space is a "triangle" with $N+1$ vertices. The $N+1$ vertex vectors of each simplex are a subset of the 2^N vertices of the hypercube they lie in. Suppose the map is known at each point on a grid. Then there is a unique way to extend the map to the interior of the simplex S whose vertices are grid points. Given a point p in the interior of S, let $\{b_i\}_{i=0}^N$ be its corresponding barycentric coordinates. Let $f(v_i)$ be the map at the ith vertex. The dynamical system at p is iterated by computing

$$\bar{f}(p) = \sum_{i=0}^{N} b_i f(v_i) \tag{6.4}$$

Linear approximation is usually used since a triangle is a linear object, but we find no serious restriction why other approximation cannot be used. Although the underlying approximating object of other approximation is not linear anymore, the topological and the dynamical properties can still be retained. Geometrically, we still have an object which looks like a polyhedron, but the faces are some curved surfaces rather than some planes. The homological and homotopic properties will not be changed by these smooth deformations.

We can then apply the approximation at each vertex, using a small neighborhood of that vertex. In this way, the dynamics is regularly approximated by a simplicial

mapping. We call this regular because the simplex constructed is based on uniform partition. There are some remarks about this approach. First, since a dynamical system is now understood by qualitative behavior and simplicial approximation can keep the topological properties of the original manifold, we know that this approximation can retain the dynamical properties of the original system. Second, this approximation requires a hugh number of training vectors to achieve a good performance. The second point is indeed a serious restriction of this approach. In both the partitioning and neighborhood construction process, many training vectors are needed to get a successful approximation. The final point is that the computation of this method is very heavy and hence the computational speed is too slow for signal processing.

To modify this approach, let us first look at a simple example. Let S^1 be the circumference of a circle of unit radius in two dimensional Euclidean space (in a dynamical sense, a limit cycle), and $x_0 \in S^1$. We first approximate S^1 by some polyhedron. A topological space which is homeomorphic to a polyhedron is said to be triangulable and the polyhedron is called a triangulation of that topological space. Since S^1 is a one dimensional space, this means finding a collection of suitably joined 1-simplexes which is homeomorphic to S^1. Pictorially we might try to open up the circle S^1 to get Fig. 6.3a.

This is not a permitted triangulation because a 1-simplex must have two distinct vertices. We remedy this in Fig. 6.3b but this is still not permitted because the 2-simplexes in Fig. 6.3b are supposed to be distinct but have identical vertices. This leads us to Fig. 6.3c which is a proper triangulation of S^1. The polyhedra K_0 associated with S^1 obtained in Fig. 6.3c, totally ordered can be written as : $K_0 = \{1\} \cup \{2\} \cup \{3\} \cup \{1,2\} \cup \{1,3\} \cup \{2,3\}$; that is, K_0 is the union of 3 0-simplexes and 3 1-simplexes (note that the fundamental group generated by this triangulation and the circle are exactly the same and are isomorphic to the group of integers under addition).

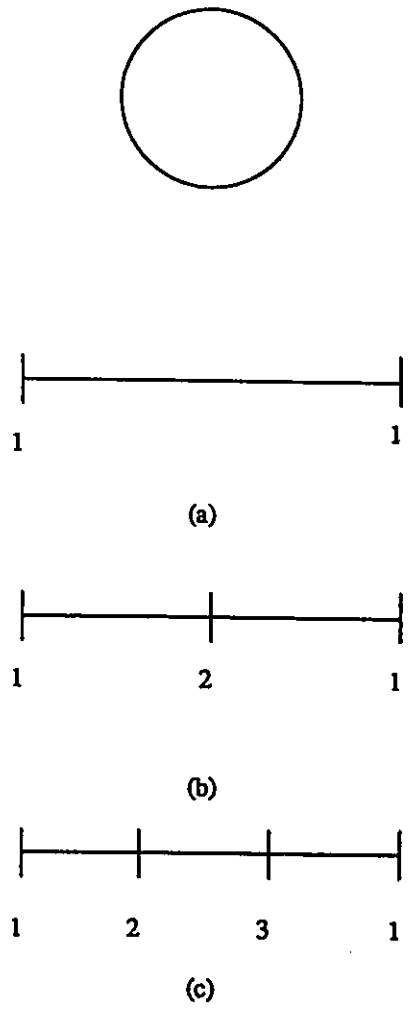


Figure 6.3: Triangulation of a circle

If we consider this as a dynamical system; that is, the attractor is a limit cycle. It is obvious that the triangulation discussed above can be applied directly. From Fig. 6.4, we see that if we partition the cycle into two pieces, the dynamics of this new structure will be totally different from the original limit cycle. If we cut the limit cycle into three or four pieces, the dynamics into each region is roughly equivalent to the original dynamics at that part of the limit cycle. If we increase the number of sections, the approximation gets better and better, till the secants match the tangents. In this case, the polyhedron is exactly the same as a circle and the dynamics is completely recovered. This example not only tells us that triangulation can approximate dynamics, but also indicates that a minimum number of vertices can be used to obtain a reasonable approximation. When a circle is approximated by two parts, the dynamics properties cannot be retained by the new geometrical objects. However, when we have three vertices, the approximation becomes acceptable. Although more vertices can give us better approximations, three vertices are indeed a crucial number in approximating a limit cycle.

When we use the embedded dynamical system, we have many points on an attractor and they are not all needed in the above sense. More clearly, we can reduce the number of the training vectors used for memorization in some way so as to reduce the computational complexity. The problem we have now is the determination of this critical number. The mathematical techniques that are usually used for this purpose are homology and homotopy theories. However, the application of these techniques to an unknown surface is quite difficult, and even harder for experimental data. In particular, we do not have too much knowledge about the sea clutter manifold; the use of these theories to sea clutter would be another challenging research problem for future study. Actually, we do not need to know the exact critical number. As long as the number of stored training vectors is greater than that number, the dynamical approximation should be acceptable. This makes us believe that the reduction of the

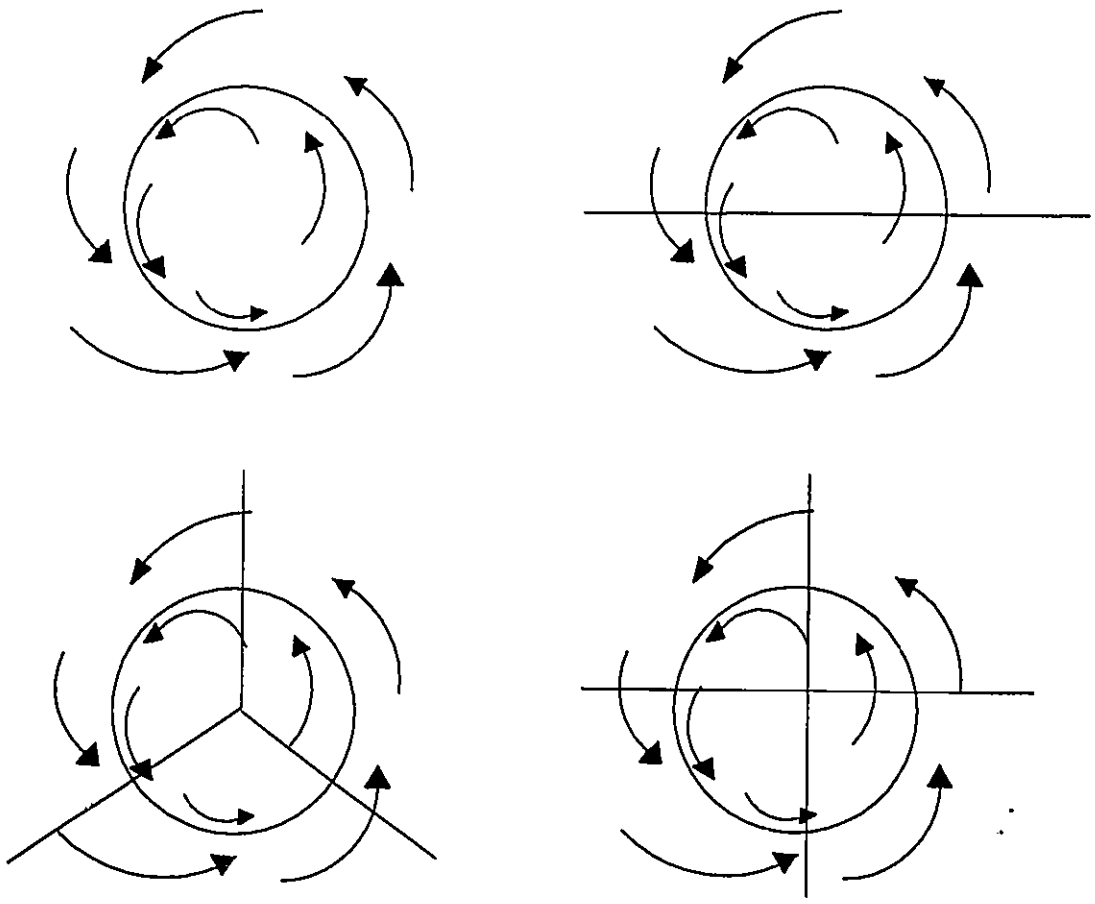


Figure 6.4: Triangulation of an attractor - limit cycle

number of training vectors is possible in our sea clutter problem. First, we have about 20,000 training vectors and it is obviously more than necessary (the critical number for a limit cycle is only 3). Second, the sea clutter manifold is not totally irregular since the dynamical attractor is strange. In the following application of the memorization method to sea clutter prediction, we will use about 2,000 to 4,000 training vectors as the memorizing database for constructing neighborhoods and charts.

An immediate question would be how do we choose these training vectors for memorization? One simple way to do this is to choose the most recently received training vectors. This approach is simple but it completely loses the information carried by those previously received training vectors. This approach is suitable for data exhibiting nonstationarity or varying statistics. Although our sea clutter is believed to be nonstationary, the stationarity period is longer than the time covered by 2,000 to 4,000 training samples. In our situation, we find that the data remain essentially stationary within 20,000 data points.

The rule we use here to find these 2,000 to 4,000 vertices is quite simple. Let $\{\mathbf{x}_i\}$ be the set of training vectors received from the radar system. i may be any positve integer and in our experiment it is usually about 20,000. Our goal is to find some vectors $\{\mathbf{v}_i\}$ called vertices (2,000 to 4,000), which can roughly represent the dynamical information carried by the whole set of training vectors (20,000). Let the number of vectors $\mathbf{v}_i$ be m. We can choose the first m vectors $\mathbf{x}_i$ from the radar to be $\mathbf{v}_i$, and then try to modify these vectors by those new inputs. We adjust these vectors by the following rule :

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \delta(n)(\mathbf{x}(n) - \mathbf{v}_i(n)) \tag{6.5}$$

where i ranges from 1 to m representing the m new vertices and n goes from 1 to 20,000 representing the total number of received inputs. At each iteration n, not all the vectrices are updated, only the one which is closest to the input at that instant

in the Euclidean sense learns that particular input. This is the well known k-mean algorithm. This algorithm can efficiently reduce the number of the vertices used in this simplicial approximation. It tries to find a new vertex which lies in the centroid of the cloud of data close to it. It is proven that this algorithm does indeed converge to the centroid. The only question is whether this algorithm will alter the actual dynamical properties or not.

Imagine that there are some points on a manifold. The points are governed by the motion equation (6.5). Since the motion is continuous, the fundamental group of the new object generated by those points are homomorphic to the original fundamental group. Topological properties are preserved (homotopy and homology), and hence we know that this simplicial approximation is equivalent in topological sense to the original simplicial approximation.

The second point about the modification is the use of a varying neighborhood. Conventional simplicial approximation uses a fixed partition of a manifold, and those vertices without given values are first approximated. This partition is similar to the suitation in constructing a neighborhood. We do not use the fixed partition to construct vertices but rather a non-uniform simplicial construction. We can think of at least two advantages. First, those points near the boundary of a fixed partition will perform very poorly. Second, the approximation at the unknown grid point may not be very good and will greatly degrade the performance.

We may now summarize what we have discussed above. First, the k-mean algorithm is used to obtain some vertices for approximation. Second, the desired values of these new vertices are approximated by some estimation techniques. The construction of a new database is then completed and hence we can apply the memory method.

Now let us apply this method to the sea clutter prediction problem. The results of the prediction of D1 are plotted in Figs. 6.5 and 6.6. Figure 6.5 uses sea clutter data

received from I-channel and Fig. 6.6 uses the Q-channel sea clutter data. Again the prediction is performed recursively; that is, the predicted value is used as the input for next prediction. The prediction is carried out for 100 iterations. It is obvious that the prediction does indeed capture the underlying dynamics.

We next extend the prediction to 500 iterations to see the performance. Figures 6.7 and 6.8 compares the prediction and the actual data using I and Q channel data respectively. The results, of course, are not as good as those plotted in Figs. 6.5 and 6.6, but it is still a reasonable prediction. The prediction can follow the trends of the sea clutter evolution.

After seeing these exciting recursive predictions results, we have two remarks that we want to make. First, all the predictions reported above are based on the zeroth order approximation; that is, averaging the desired response of the training vectors. In principle, if the data are generated from a deterministic rule, then higher order approximations should give a better performance. We have applied the first order approximation by using singular value decomposition to data set D1, and the comparsion of mean squares prediction error is plotted in Fig. 6.9 which obviously performs better than the averaging method. When we go to the second order case, we find that special attention must be paid to obtain a reasonable prediction due to the robustness problem mentioned earlier in this chapter. We therefore recommend the use of the zeroth and first order prediction for the sea clutter data we have, and based on the experience from computer experiments, we prefer the zeroth order method to the first order because of its efficiency.

Second, *Takens Embedding Theorem* has a very important implication that measuring any experimental observable would not produce any loss in information in the qualitative sense of a dynamical system. In other words, we should obtain a similar result by using either amplitude, in-phase component or quadrature component. We perform comparisons of recursive prediction errors of data sets D2 to D5 using

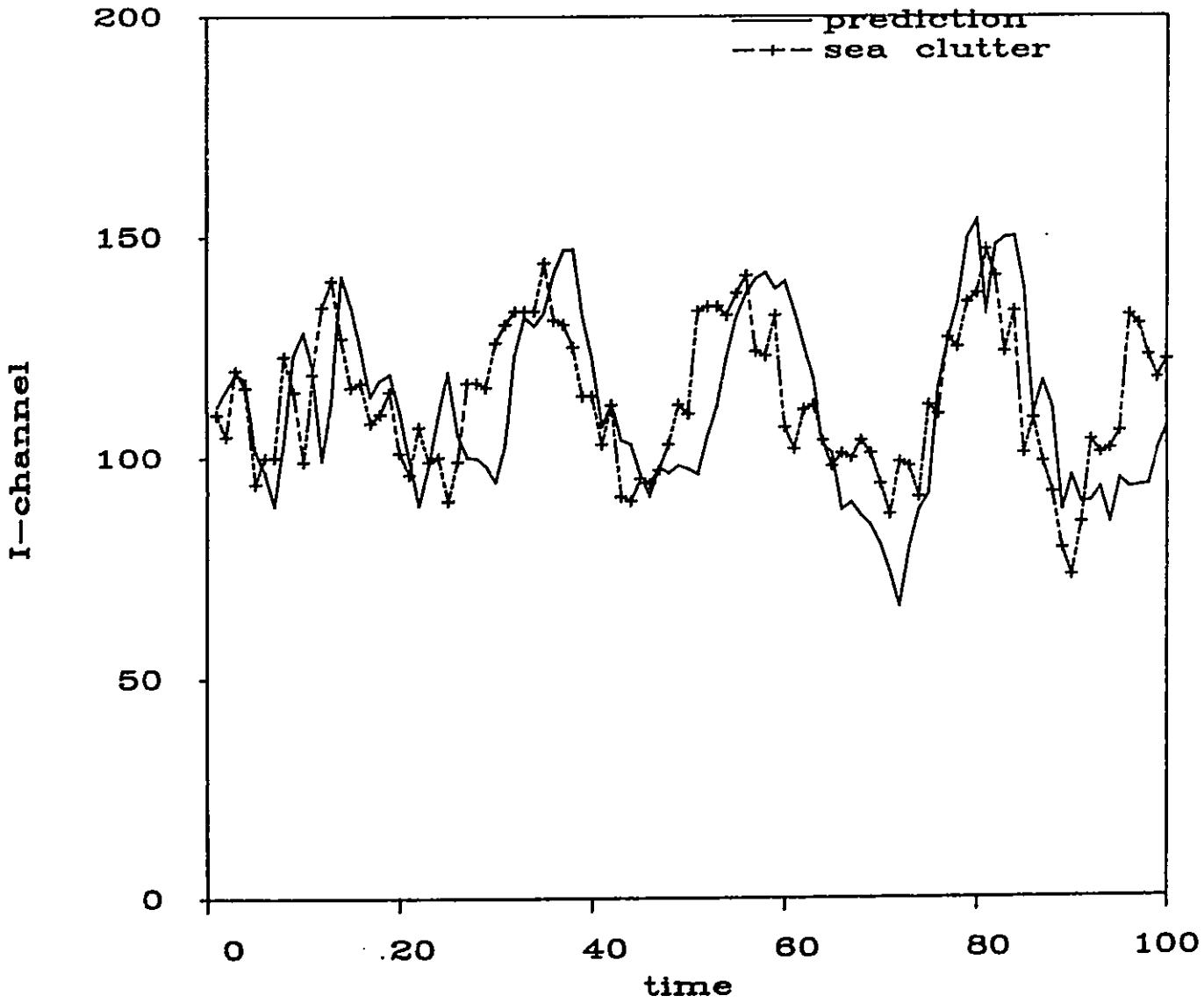


Figure 6.5: Prediction of sea clutter using I-channel D1 data (100 iterations)

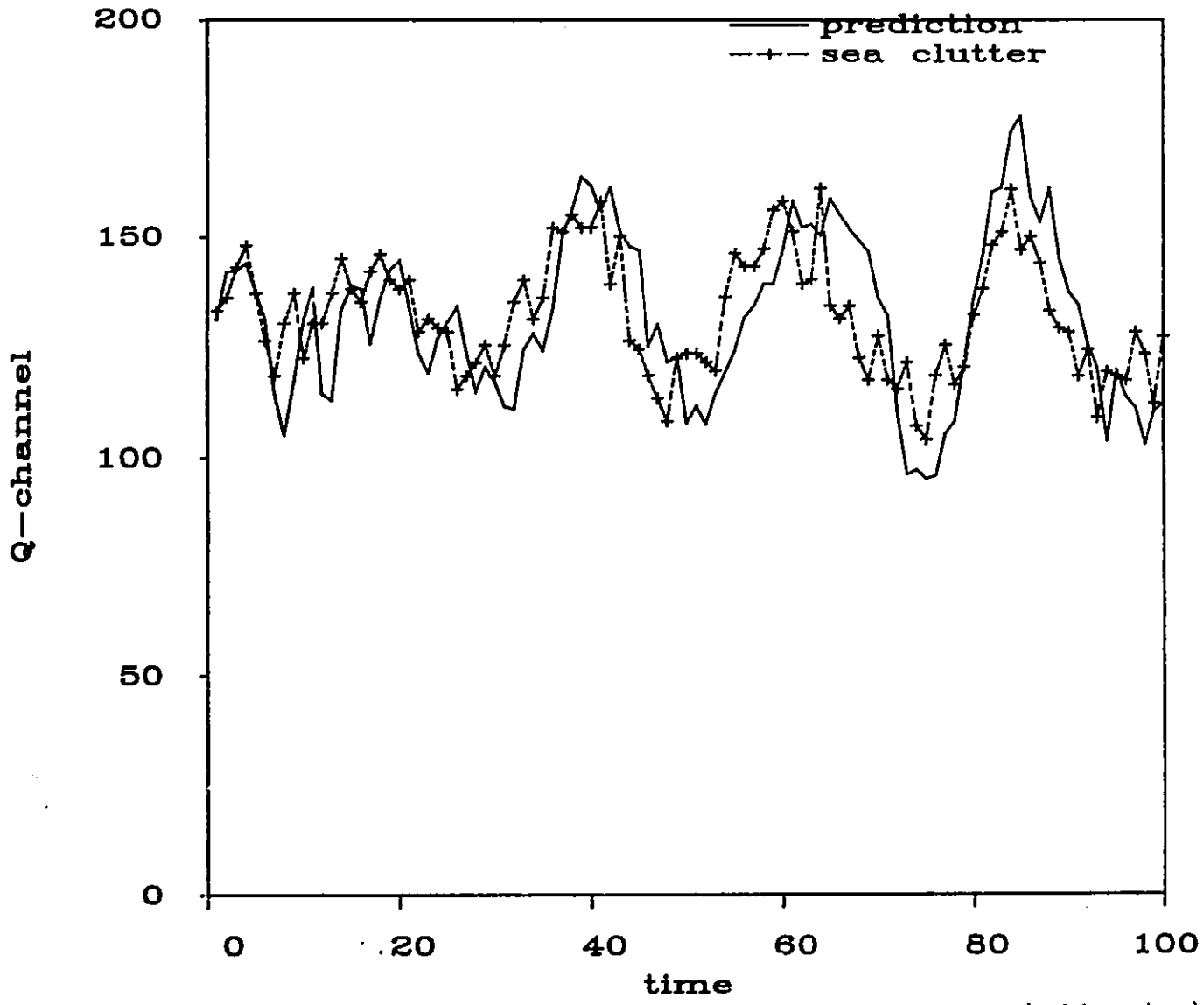


Figure 6.6: Prediction of sea clutter using Q-channel D1 data (100 iterations)

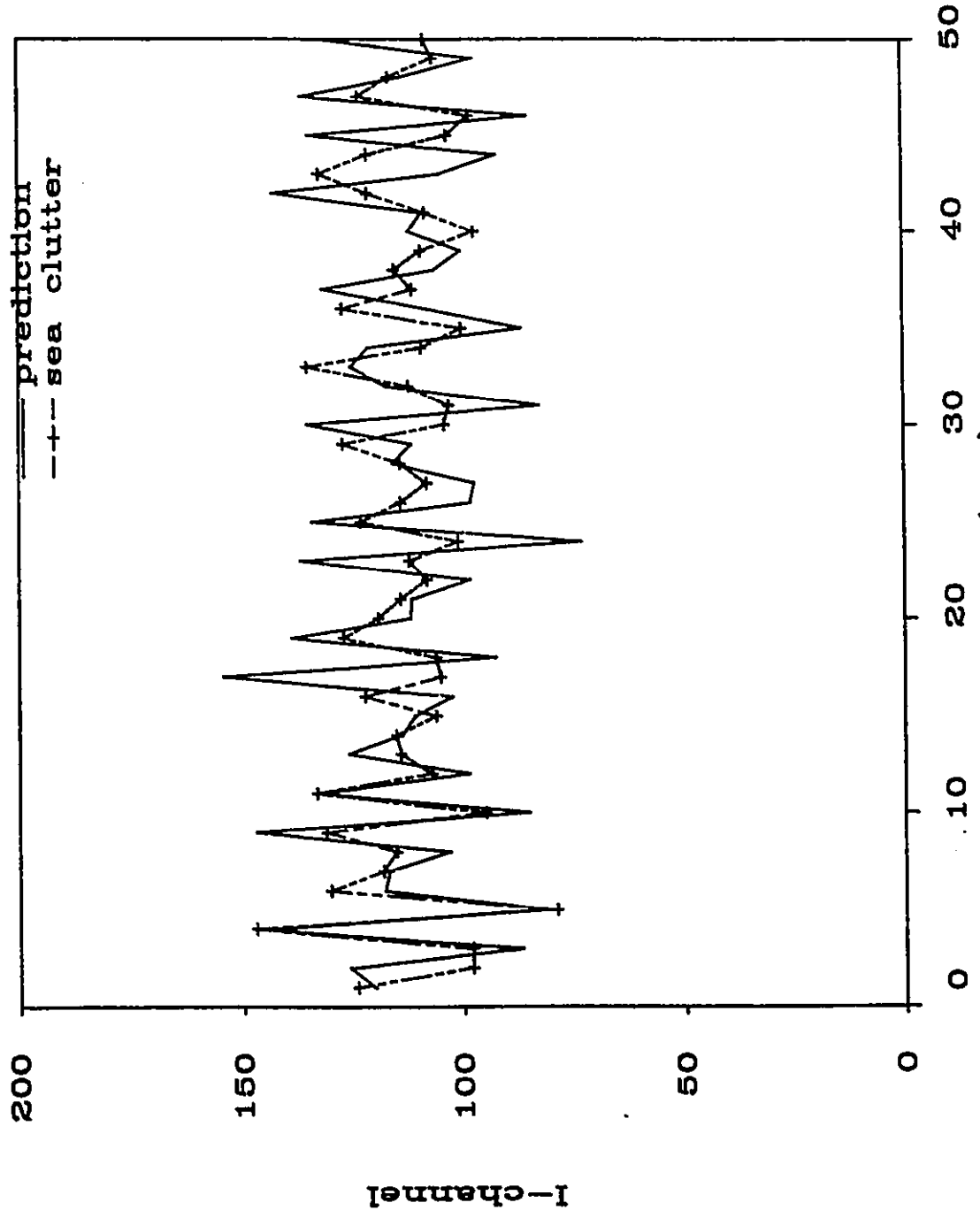


Figure 6.7: Prediction of sea clutter using I-channel D1 data (500 iterations)

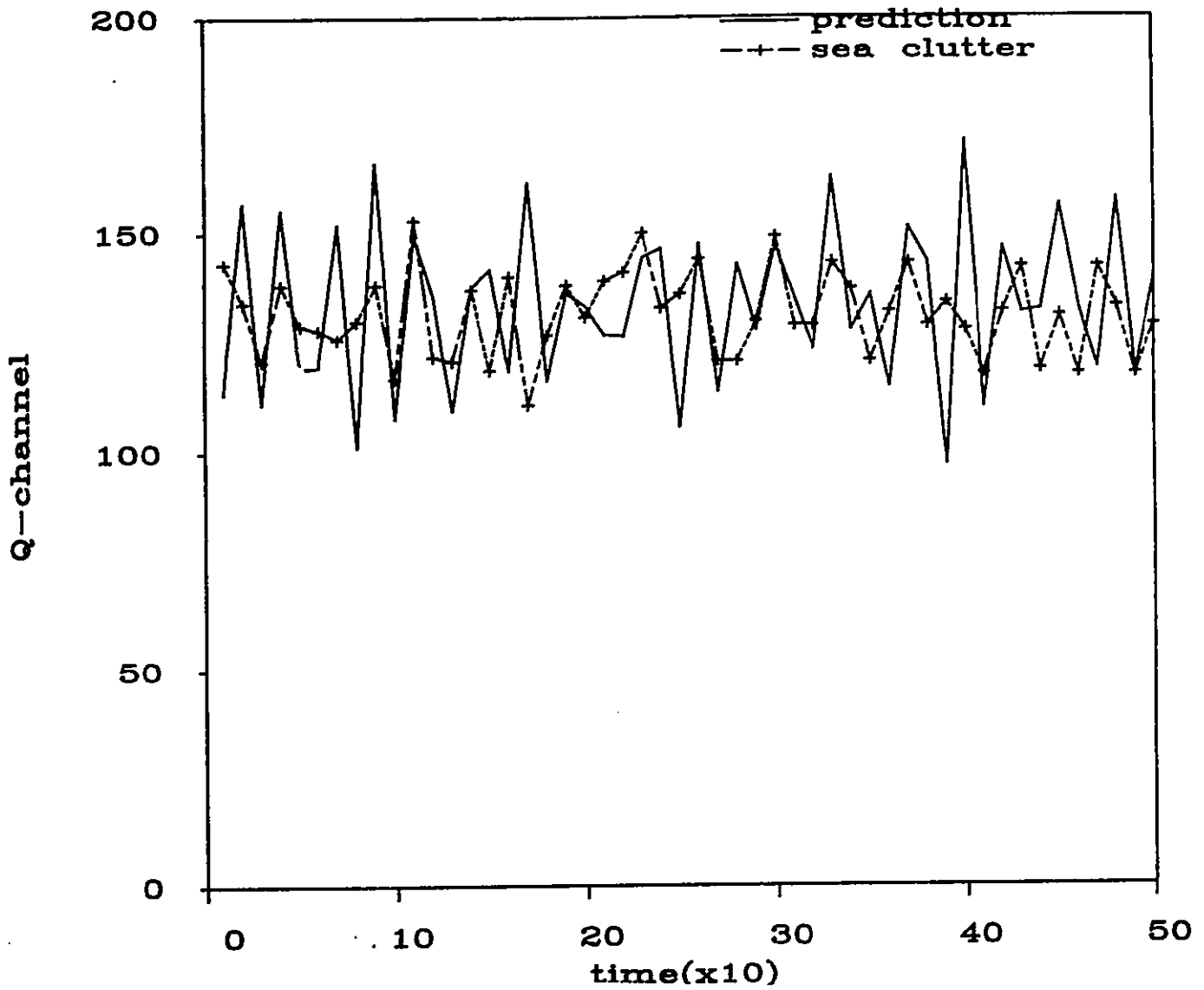


Figure 6.8: Prediction of sea clutter using Q-channel D1 data (500 iterations)

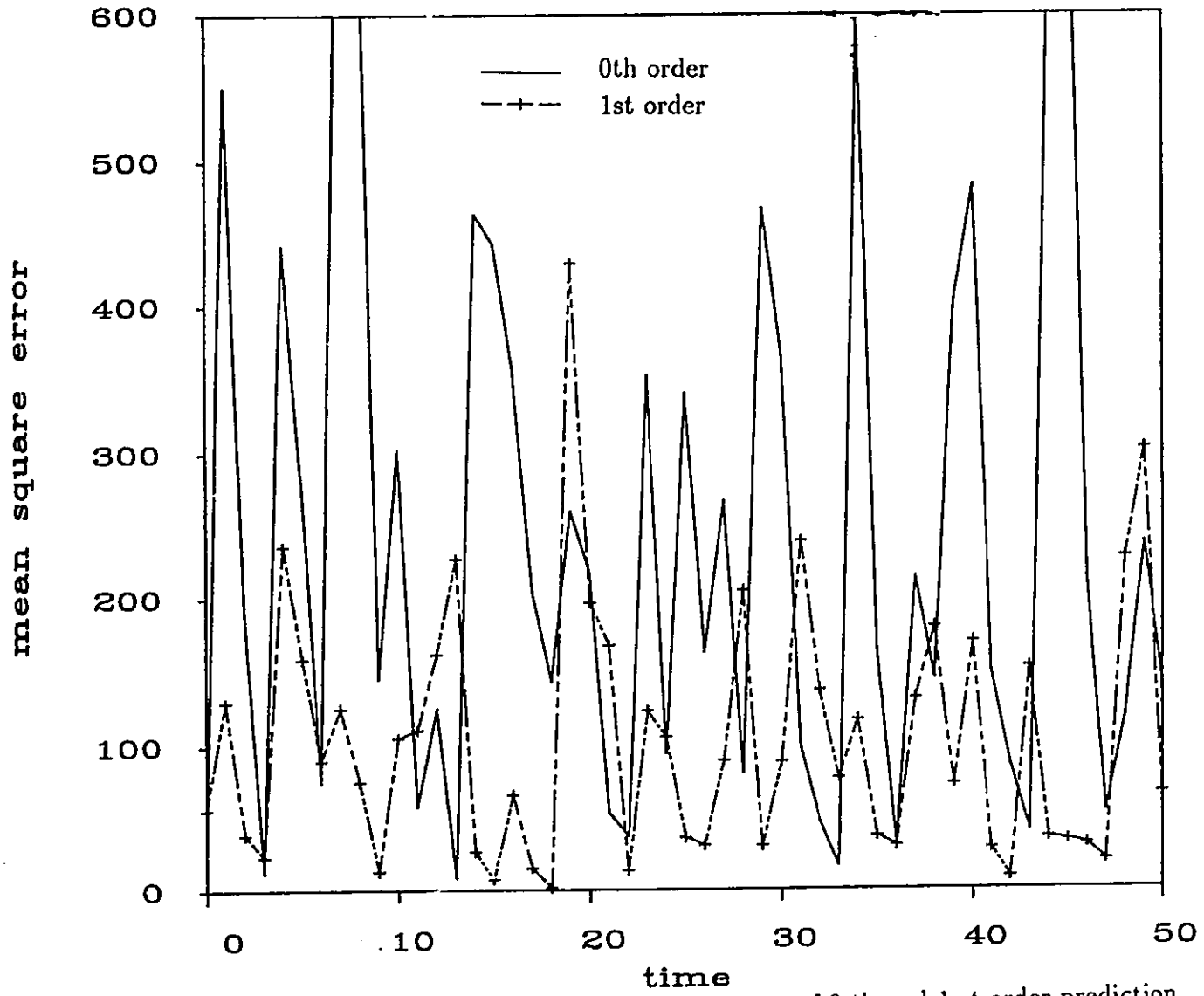


Fig. 6.9 Comparison of prediction error of 0-th and 1-st order prediction

mean value of D1 ~ 130

these three observables. The prediction is based on the zeroth order method, and the results are plotted in Figs. 6.10 to 6.13.

We also take the average of the errors of these one hundred iterations, and the results are listed in Table 6.1. We can see that the prediction errors based on different input observables are indeed very closed. This observation confirms the importance of this single observable measurement idea which we have explaind more clearly in Chapter 2.

6.4 Chaotic Detection of Small Targets in Sea Clutter

So far we have only discussed the modeling of sea clutter. The final objective is to apply this model to practical problems like detection. The problem of detecting small targets in sea clutter is a very challenging and practical problem. By a small target we mean a small boat, a growler (small piece of ice) or a waverider; in each of these cases, the radar cross section is small. Conventional detection schemes applied to this situation do not seem to be successful. Therefore, it is our intention to apply the chaotic approach to this problem. Before doing that, let us briefly review the basic detection theory, and how chaos can be introduced into this classical technique.

In a radar system whose soie purpose is to detect the presence or absence of a target, the signal set consists of only elements, $s_1(t)$, corresponding to target present, and $s_0(t)$, corresponding to target not present. Because of factors such as the randomness of the disturbances in the channel, it is believed that there is in general no deterministic mapping from the elements of the signal space to the observation space. This randomness is dictated by the statistics of the channel noise. Hence, in the classical detection theory, it is believed that for a given element in the signal

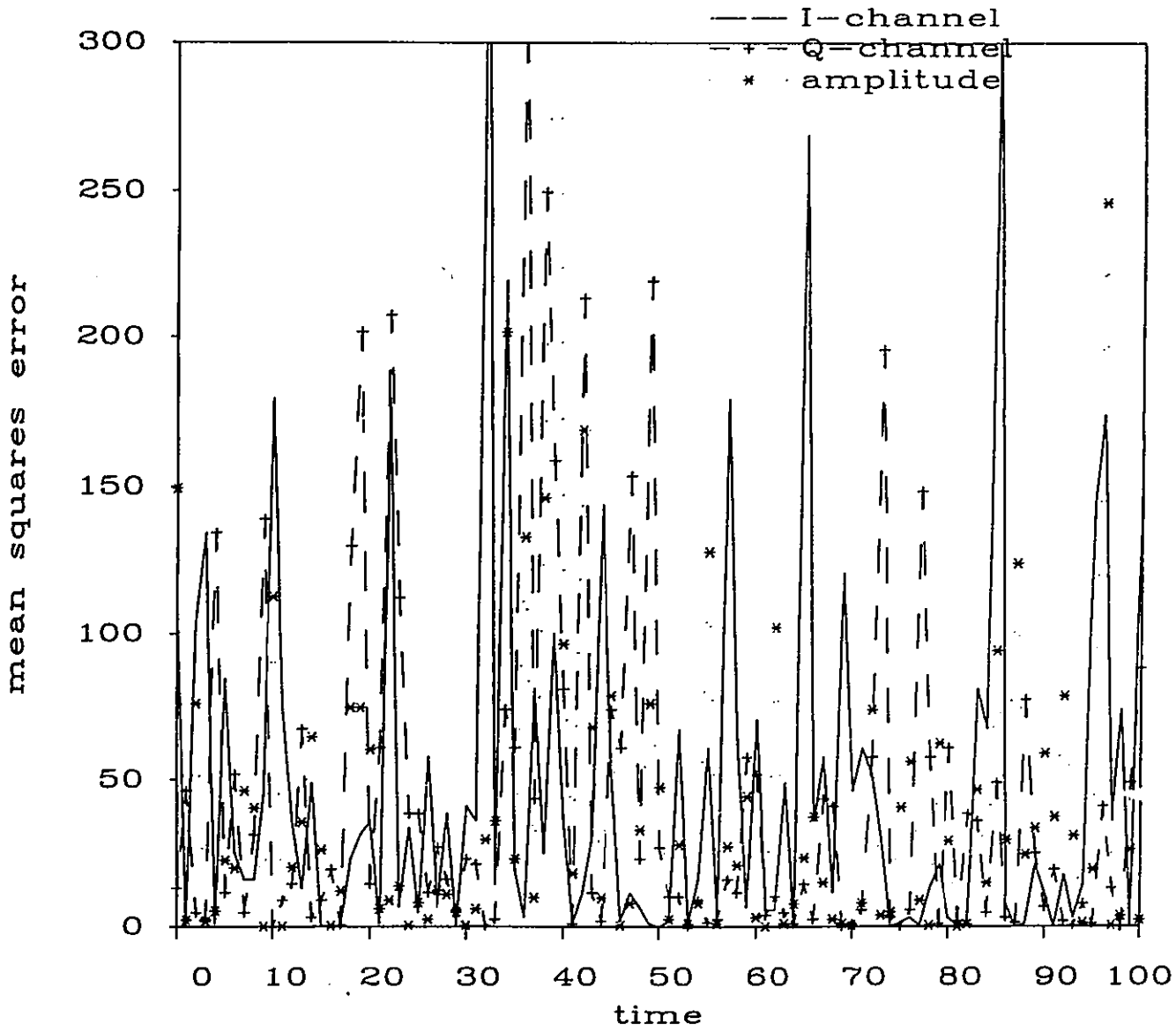


Figure 6.10: Comparison of prediction error of D2 using amplitude, I and Q as input (mean value of D2 $\sim$ 130)

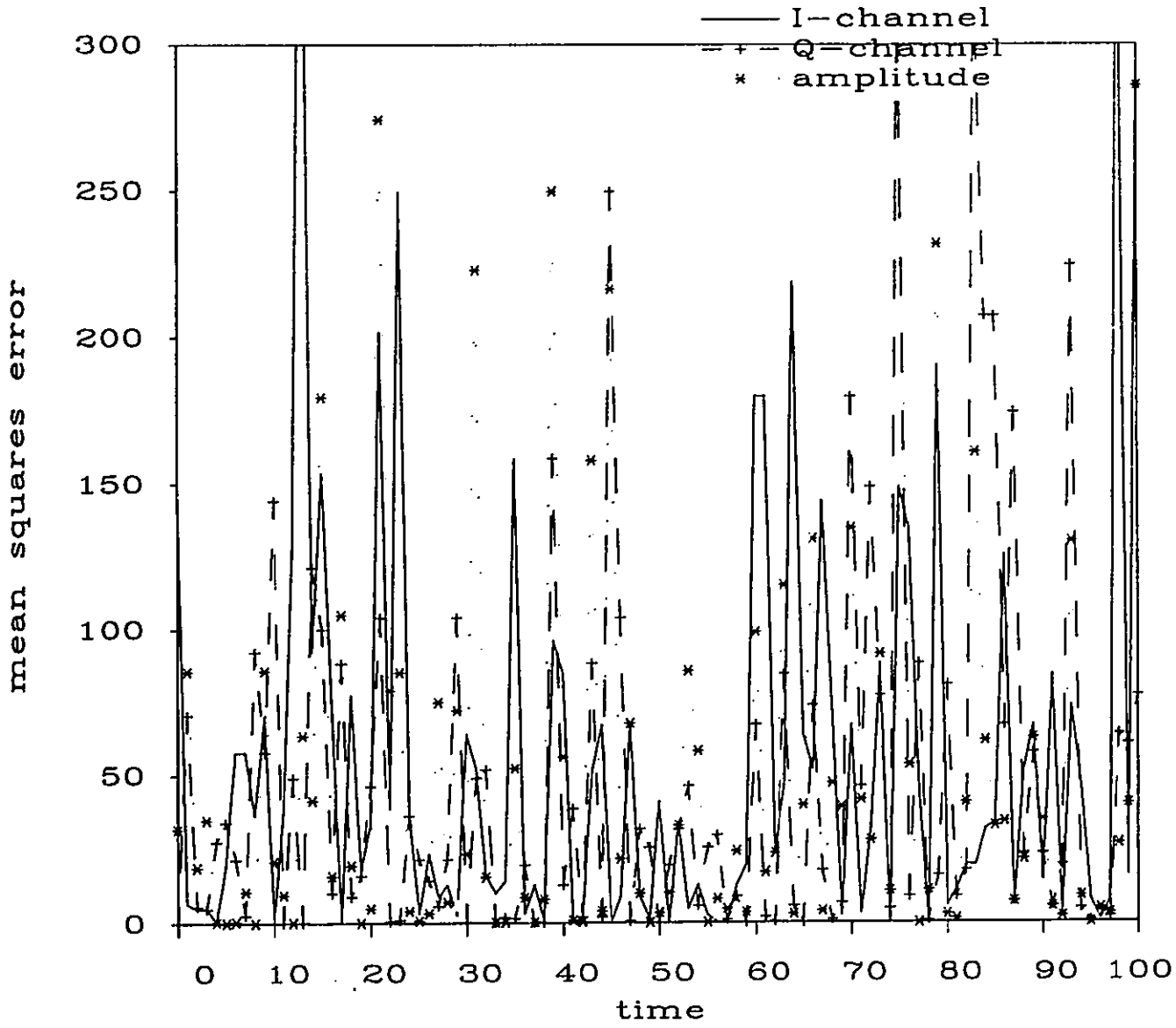


Figure 6.11: Comparison of prediction error of D3 using amplitude, I and ! as input (mean value of D3 ~ 130)

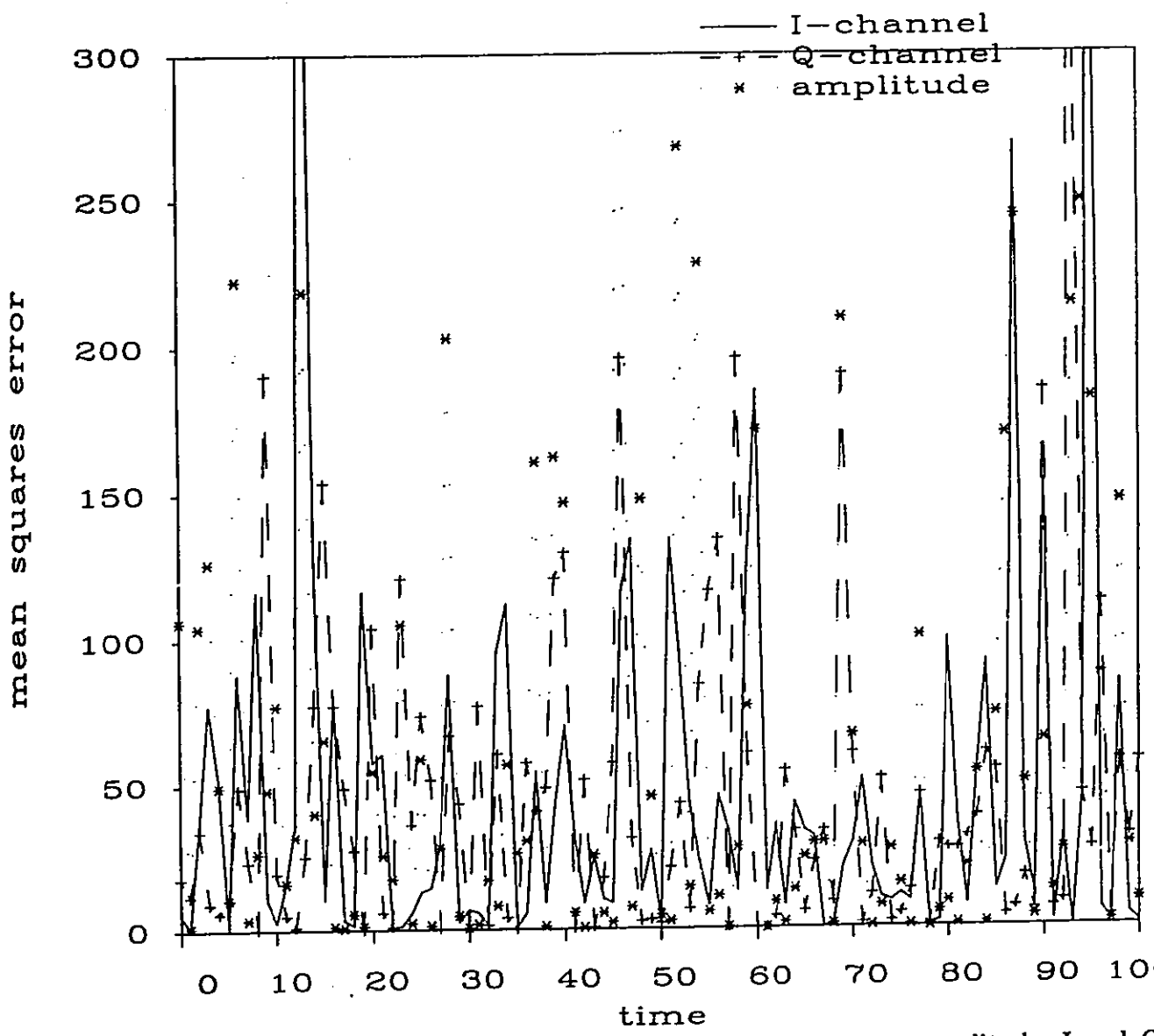


Figure 6.12: Comparison of prediction error of D4 using amplitude, I and Q as input (mean value of D4 ~ 130)

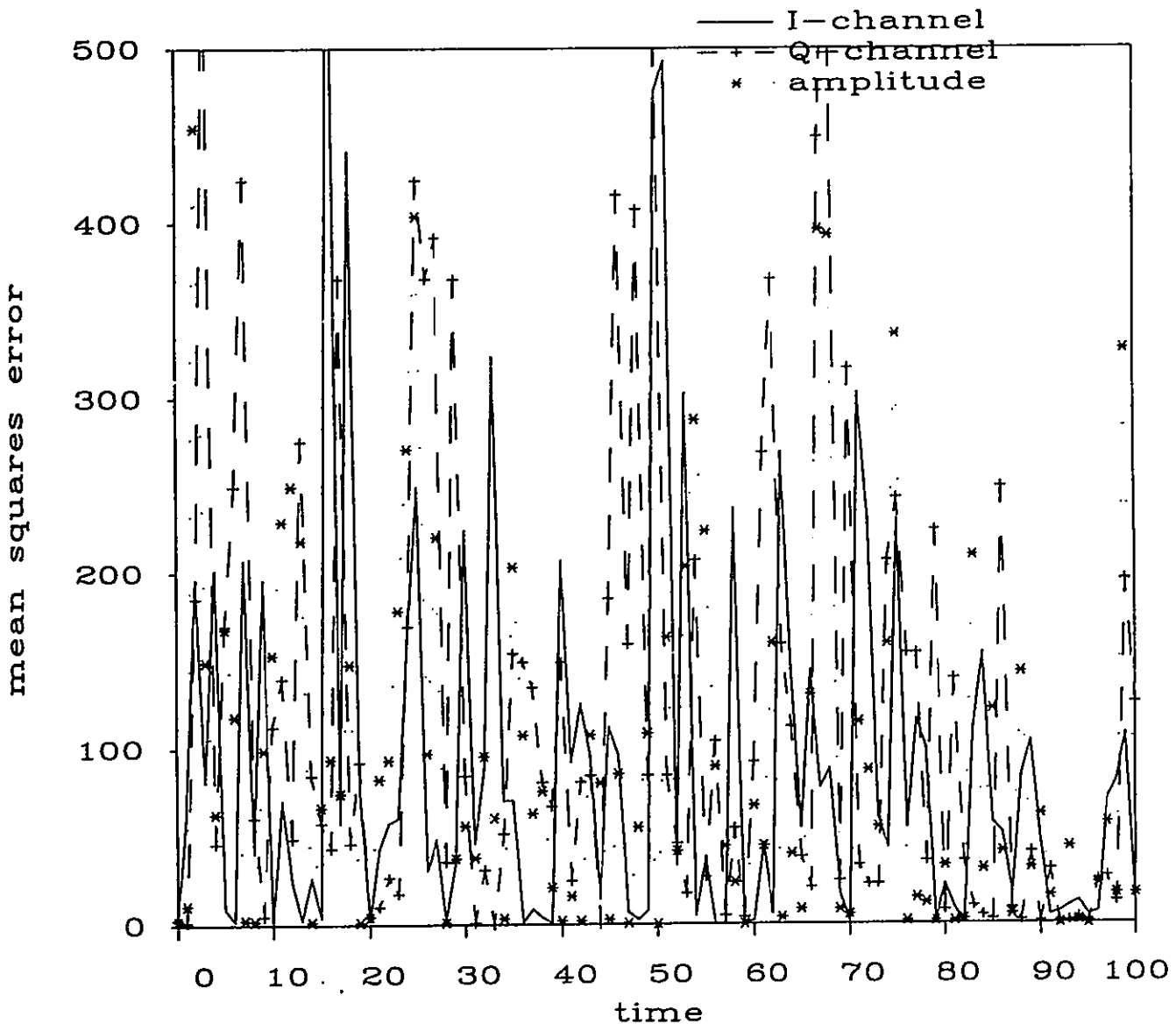


Figure 6.13: Comparsion of prediction error of D5 using amplitude, I and Q as input (mean value of D5 ~ 130)

D2	mean squares error	D3	mean squares error
I-channel	48.39086	I-channel	58.12121
Q-channel	44.95568	Q-channel	49.72115
Amplitude	41.22416	Amplitude	50.65950
D4	mean squares error	D5	mean squares error
I-channel	48.50023	I-channel	91.16909
Q-channel	46.33153	Q-channel	126.0261
Amplitude	59.12655	Amplitude	91.03349

mean value ~ 130

mean squares value $\sim 1.7 \times 10^4$

Table 6.1: Comparison of averaged prediction error using amplitude, I and Q as input

space the values that observed data may assume can at best be represented probabilistically. Putting this in mathematical lanaguage, detection is formulated into a statistical decision problem on the two hypotheses:

$$\begin{aligned} H_1 &: \quad s(t) + n(t) \\ H_0 &: \quad n(t) \end{aligned} \tag{6.6}$$

where $s(t)$ and $n(t)$ represent the signal and noise processes at time t respectively. Classical detection is based on the statistics of the noise process (more precisely, we use the probability density of $n(t)$, and then obtain a likelihood to determine which hypothesis is more likely).

This detection is developed on the assumption that random behavior must be described by statistical methods. Now we know that chaos theory has strongly shaken this assumption, and we may reformulate the radar detection problem in a totally new fashion. The main difference is that the noise process $n(t)$ is not necessarily a random process, which can be described by a nonlinear dynamical system with the ability to generate any kind of noise, white, colour, or wide-band. Detection can be viewed as a process of distinguishing two dynamical systems rather than hypthoeses testing using statistics. Now the question is: How do we distinguish two different dynamical systems?

It is supposed to be a simple question. If we have two dynamical systems, say H_0 and H_1, trajectory from system H_0 cannot match to the trajectory from H_1 and vice versa. For instance, it is very easy to see that a periodic curve cannot be matched to a quasiperiodic motion. There are many ways to distinguish two dynamical systems. For example, Fourier transform techniques would be good enough to identify the systems in above simple example. For chaotic system, we may use dimension, divergence or many other invariants. In this thesis, we would like to use the basic idea of "matching trajectories"; that is, trajectories from different dynamical

systems do not match. In other words, if we can somehow obtain the dynamics of one system (for example, noise only), and fit the trajectory or data from the other system (noise plus signal) to the first system. A very "unfitted" situation will come up, and we can use it to tell which system the trajectory should belong to. We are now going to explain this idea more clearly in terms of radar detection of small targets in sea clutter.

In order to establish a framework for detection, a target can be viewed as an object embedded in sea clutter. When the electromagnetic wave from the radar hits this object, the dynamical property or the statistics of this return is expected to be different from those scattered by an ocean surface. In our problem, no property of the background (ocean surface) is assumed. Therefore, the detector must learn from the environment to identify the existence of a target. This requires the use of an adaptive detection. This problem is similar to the problem of detecting small regions of an image which differ from their surroundings in areas such as optical aerial reconnaissance, radar imaging, and medical diagonosis through imagery.

The basic procedure for performing this adaptive detection can be described by the following steps. First, we choose a model. Second, this model is exposed to the environment to learn the underlying statistics or dynamics. Third, we freeze the model and use it as a detector with new entries to the model. Since the underlying processes for target and ocean surface are different, we will observe an abrupt change in the model behavior when new data representing a target are fitted to the model. One may question whether the back scattering from targets and ocean surface are indeed different or not. We think the answer is positive, based on the success of classical adaptive detection or clutter suppression techniques, although there is no strict theoretical physics or mathematics to support this assertion.

Our detection derives from the fact that a constant false alarm rate (CFAR) significance test can be transformed to a test involving error residuals of an adaptive

predictor. In other words, a small prediction error implies a good detector. We choose a threshold based on the prediction error from the learning data and use it as a detection criterion. Basically, our detection is very similar to a conventional adaptive detection scheme. The main difference is the replacement of linear prediction by nonlinear prediction.

The background data used for the detection are D4 and D5. The target for D4 and D5 is a sparbouy and a waverider, respectively. The sparbouy is a very strong target. Thus, when the data received contains this target, the prediction error will immediately appear to be very large. We report this observation in Tables 6.2 and 6.3. Table 6.2 contains the square of the prediction error when the trained predictor meets some data which consist of backscattering from the sea surface alone. The prediction errors are pretty small. The magnitude of the data lies between 0 and 255, and the mean value is about one hundred thirty. The mean square prediction error of sea clutter data has a mean of about one hundred twenty. The occurrence of a mean square value of prediction error greater than 600 is extremely rare. Table 6.3 illustrates the prediction error for backscattering from the sparbouy. The mean square prediction error is much larger than those listed in Table 6.2. If we choose the mean square error equal to 1000 as the threshold, the existence of a target is easily identified.

Tables 6.4 and 6.5 contain results similar to Tables 6.2 and 6.3, except the background clutter is D5 and the target is a waverider. This is the most difficult target we have in the data collected at the field trip. It is difficult to detect because the target is too small and hence it is often blocked by the presence of sea waves. So even though the radar is pointed directly at the target, most of the data received is just sea clutter data. In view of this fact, we have to choose a threshold by using the training data and check the data received by direct target pointing to see whether any portion of the data sequence has an abrupt change of behavior or not.

i	prediction error	i	prediction error
1	69.77770	51	7.222205
2	20.11108	52	8.888862
3	278.7778	53	527.2220
4	205.0000	54	25.88894
5	9.444421	55	58.88885
6	27.22218	56	115.2222
7	49.44445	57	59.22230
8	37.77776	58	57.77776
9	15.22218	59	82.77779
10	193.7776	60	139.7777
11	46.22218	61	307.3557
12	74.88882	62	222.7777
13	28 88889	63	43.77780
14	367.2221	64	29.88886
15	149.0000	65	4.444441
16	5.555564	66	57.77782
17	92.11108	67	217.8889
18	6.777770	68	124.8889
19	22.22220	69	72.55550
20	81.11111	70	187.7778
21	43.77780	71	25.11111
22	100.1111	72	7.222205
23	44.44447	73	41.88882
24	43.77780	74	129.7779
25	5.777785	75	26.77778
26	145.8890	76	93.44441
27	180.5556	77	94.77785
28	22.44448	78	151.1111
29	32.22228	79	4.000000
30	29.88886	80	59.22216
31	21.44446	81	113.4445
32	86.44446	82	62.44448
33	80.55562	83	128.5556
34	54.77780	84	182.2222
35	75.22218	85	85.88898
36	38.44443	86	154.8838
37	18.00000	87	49.44448
38	46.77776	88	473.1112
39	26.77778	89	43.22216
40	108.5555	90	36.11111
41	325.5554	91	277.5555
42	122.0000	92	32.00000
43	45.55550	93	47.22228
44	121.4444	94	284.8890
45	165.5557	95	329.4444
46	1.888897	96	548.4446
47	250.7777	97	87.22213
48	78.44449	98	49.88883
49	37.44447	99	135.2222
50	82.77773	100	50.77779

Table 6.2: Prediction error of D4

i	prediction error	i	prediction error
1	10800.11	51	7034.000
2	11025.00	52	7570.000
3	10370.00	53	8177.000
4	9257.000	54	10588.89
5	10369.00	55	11432.22
6	10485.00	56	10118.44
7	8765.000	57	8709.444
8	10116.00	58	7825.000
9	8244.000	59	7397.000
10	8146.000	60	9616.000
11	7738.000	61	5378.778
12	11317.89	62	6553.445
13	8685.000	63	6691.556
14	9797.778	64	7218.888
15	10760.00	65	7297.000
16	8594.000	66	10389.78
17	10992.22	67	6963.777
18	8320.000	68	7905.890
19	8721.444	69	7298.000
20	7361.000	70	7367.223
21	7457.000	71	6805.000
22	8273.890	72	9081.000
23	6772.000	73	8820.000
24	7298.889	74	6505.000
25	6025.888	75	7804.111
26	8780.556	76	6724.000
27	10610.00	77	7261.000
28	12273.45	78	7156.000
29	11833.00	79	9424.557
30	11765.00	80	9189.000
31	7555.777	81	10484.00
32	8675.777	82	8009.000
33	9540.000	83	8825.000
34	11956.00	84	9565.000
35	9810.000	85	6501.557
36	9896.000	86	6055.557
37	10673.00	87	6221.000
38	11077.45	88	5320.888
39	9614.224	89	6073.888
40	8792.557	90	7413.555
41	11355.11	91	6697.888
42	11890.00	92	7696.555
43	9034.000	93	9026.000
44	10422.78	94	10322.00
45	8545.000	95	7289.000
46	9105.778	96	6909.444
47	9006.222	97	7973.556
48	7978.000	98	8410.000
49	7445.000	99	8874.000
50	7345.000	100	9872.000

Table 6.3: Prediction error of sparbouy in D4 environment

Table 6.4 records some typical prediction error of sea clutter data. In our computer experiment, we find that the mean square prediction error seldom goes beyond 600 and almost never becomes greater than 700. We therefore choose 1000 as the threshold and detect whether some part of the target data sequence has an error larger than this threshold. The result is depicted in Table 6.5. We use 5000 data points for the experiment which is approximately 25 seconds long, and we find that a certain number of points in the data do exceed the threshold.

In this chapter, we first introduce the idea of using memory to construct a neural network rather than using learning. Surprisingly, this idea has a very strong mathematical foundation in differential topology. Based on differential topology, we have a deeper understanding about the sea clutter manifold, and obtain a new way to perform filtering and prediction by treating data as a manifold rather than a set of random variables. We further extend this dynamics and manifold idea to detection, and derive a deterministic approach to detection problem. The method is more powerful than those based on probability density function, and we apply this method to detect some small targets in sea clutter.

i	prediction error	i	prediction error
1	17.77776	51	0.4444512
2	6.777770	52	3.777795
3	76.55563	53	82.77779
4	18.22227	54	0.0000E+00
5	4.444441	55	22.44448
6	8.888923	56	47.11108
7	139.7777	57	32.44447
8	68.00000	58	127.2222
9	24.55561	59	40.55560
10	233.0000	60	30.22223
11	152.5555	61	28.88883
12	74.77785	62	16.55555
13	47.22215	63	89.00000
14	455.2225	64	53.44437
15	3.777795	65	128.8888
16	5.444421	66	128.8888
17	249.4445	67	145.0000
18	17.00000	68	34.00000
19	50.88897	69	1.888877
20	98.22223	70	89.88895
21	1.777764	71	100.1111
22	105.4445	72	10.88888
23	87.22213	73	130.0000
24	1.888892	74	227.2221
25	13.88892	75	87.22213
26	13.00000	76	25.11111
27	168.1112	77	19.77773
28	305.1112	78	43.55558
29	12.88889	79	418.8890
30	180.5556	80	169.0000
31	16.1111	81	49.88883
32	156.5555	82	28.88892
33	43.11114	83	51.22218
34	2.222236	84	60.55564
35	112.2223	85	547.7779
36	65.88888	86	22.22219
37	16.00000	87	12.55552
38	88.11116	88	74.88882
39	95.22233	89	75.55552
40	241.7778	90	45.00000
41	94.44443	91	0.111112
42	25.88894	92	41.44442
43	61.00000	93	34.88888
44	144.0000	94	61.88880
45	228.1112	95	101.7778
46	123.2222	96	153.1112
47	221.0000	97	280.1112
48	42.88882	98	51.77777
49	82.88879	99	70.77776
50	86.22215	100	85.77782

Table 6.4: Prediction error of D5

i	prediction error
304	2348.444
655	1040.444
673	1485.445
2025	1227.222
2503	1058.889
2631	1110.222
2643	1519.222
3432	1381.445
3713	1808.444
3776	1045.889
4242	1024.111

Table 6.5: Prediction error of waverider in D5 environment

Chapter 7

Discussion and Conclusion

Radar clutter has a long history of being modeled as a stochastic process. Surprisingly, we cannot find any research work in the literature which tries to justify the validity of this fundamental assumption. More precisely, nobody has attempted to prove or disprove the argument that radar clutter is indeed random. What most people try to do is to find a better stochastic model than the existing one. "Better" may be closer to some physical phenomenon, or matching closer to experimental data. It seems to us that the assumption that radar returns come from a purely random process has been taken for granted. It has been our intention to consider this fundamental issue in this thesis.

What we have attempted to do here is to challenge the premise that radar returns are samples of a stochastic process. We suggest that although radar returns have a highly random appearance, they do not imply that the random appearance must be the result of a purely random process. The mathematical theory which backs up our thinking is the recent breakthrough in dynamics; namely, deterministic chaos. More precisely, the random appearance of radar returns may be the result of a chaotic phenomenon. This is a very important idea, since if a process is deterministic and we

model it by statistical method, useful information will be destroyed. Therefore, we need to show whether radar returns can be described by chaos or not.

Our chaotic analysis is based on experimental radar data rather than scattering theory. This is the conventional modeling procedure used in radar. Moreover, being based on real-life data, the results of our chaotic analysis should be of more practical value to engineers. Two conventional techniques, dimension and local divergence, are performed on five different sea clutter data sets. Both analyses on all five data sets point to the direct[illegible] that modeling sea clutter as a deterministic model is an appropriate method.

To get a model or extract the underlying deterministic dynamics of sea clutter, we consider the inverse problem as a problem of inductive inference. The modeling problem can then be viewed as a problem of prediction by learning from the environment. The use of a neural network provides a powerful tool in solving the prediction/modeling problem.

We first tried to use conventional neural networks such as the backpropagation network, and found that they are not suitable for our problem. The major difficulty in the use of the backpropagation algorithm is that it is not designed for real-time applications. A novel neural network structure based on rational functions is proposed to solve our modeling problem. This network has a unique local minimum, and hence has a rapid learning speed. Moreover, it does not require repeated training as the backpropagation. We also find that it can model the sea clutter dynamics by a low complexity structure, and the performance is quite good.

We have also attempted to model sea clutter by another neural network based on the idea of memorization. This approach uses the result that sea clutter can form a manifold in Euclidean space, and hence the complicated filtering problem is converted to a linear filtering or simple averaging process on a manifold. The prediction of sea clutter evolution using this method is very good and robust. From our experiment,

we find that this chaotic adaptive filtering technique is even better than the rational function neural network with respect to both performance and robustness. Thus, we have used this method to detect small targets in sea clutter to see whether this dynamical modeling approach is useful or not.

Our detection method is totally different from the conventional ones. Conventional detectors use the idea of probability distributions and likelihood functions to perform decision on the presence or absence of targets. In our theory, the two hypotheses are no longer just two random processes which can only be described by probability distributions. Instead, they are two different dynamical systems. Detection becomes a process of distinguishing dynamical systems. We apply this idea to our real-life data, and we have demonstrated that the method can indeed detect small targets in the presence of sea clutter.

Aside from engineering applications, a predictive model also serves the purpose of a chaotic test for sea clutter. The results of this prediction analysis tell us that prediction can indeed capture the motion of sea clutter. This is another strong evidence that sea clutter is not purely random.

Sea clutter is known to have a continuous power spectrum with a strong low-frequency content. This, therefore, rules out the modeling of sea clutter by a fixed point, periodic motion, or quasi-periodic evolution. At present, we only know of five possible models for a waveform: fixed point, periodic, quasi-periodic, random, and chaotic. We have ruled out the applicability of the first four models. This therefore suggests the possible use of a chaotic model as a mathematical descriptor of sea clutter, which is the premise of this thesis.

In order to be definite on whether sea clutter is chaotic, we recommend that additional work be undertaken as follows:

1. More sea clutter data sets in different environments, and longer data sequences

should be used to obtain more convincing results. Also, other tests such as the generalized dimension analysis [32] and unstable periodic orbit analysis [2], which have been demonstrated to be quite powerful, should be applied to test the sea clutter data.

2. A theoretical chaotic scattering model should be established. The basic idea of the model is as described in Section 1. The main point is to combine the chaotic scattering theory [11] with the existing scattering model for sea clutter, that is, K-distribution. We want to stress the point that the K-distribution is based on a random walk model which is obviously different from our approach. However, this model gives us a clue that sea clutter may be considered as consisting of a few scattering centers; this is exactly the same situation for a chaotic scattering model described in [11]. We strongly recommend the development of this theoretical approach to further substantiate the chaotic modeling of sea clutter.

To summarize the thesis, our main idea is to apply nonlinear dynamics or chaos theory to the theory of signal processing. Conventionally, signal processing problems are solved by statistical methods. Now, we know that chaos theory can model behavior that appears to be random, and it provides a deterministic treatment of the problem. We therefore have an idea of replacing the role of statistics in signal processing by chaos theory. We coin the term "*chaotic signal processing*" for this novel idea in signal processing.

We first tested our idea in chaotic signal processing by using some real-life radar data. Sea clutter, a process usually modeled by probability distribution, has been the basis of our first experiment. We have found that chaos theory provides a useful tool in describing sea clutter. We then developed signal processing techniques such as detection, filtering, prediction and modeling based on a deterministic approach

rather than the conventional statistical method. These techniques not only have very profound and interesting theories to support them, but they are also very promising in their applications to real-life situations.

To conclude, we find the idea of chaotic signal processing to be both theoretically interesting and practically useful. Indeed, the use of chaos in signal procssing has a lot of potential. However, we stress the need for more extensive research on chaotic signal processing, in both theoretical and experimental terms, to make it fully convincing.

References

1. Abraham,N.B., A.M.Albano, B.Das, G.De Guzman, S.Yong, R.S.Gioggia and J.R.Tredicce, "Calculating the dimension of attractors from small data sets," Physics Letters, Vol.114A, No.5, pp.217-221, 1986

2. Auerbach,D., P.Cvitanovic, J-P.Eckmann, G.Gunaratre, and I.Procaccia, "Exploring chaotic motion through periodic orbits," Physical Review Letters, Vol.58, No.23, pp.2387-2389, 1987

3. Badii,R. and A.Politi, "On the fractal dimension of filtered chaotic signals," in *Dimensions and entropies in chaotic systems*, edited by G.Mayer-Kress, Springer Verlag, New York, 1986

4. Barron,A.R. and R.L.Barron, "Statistical learning networks : a unifying view," Computing Science and Statistics : Proceedings of the 20th Symposium on the Interface, pp.192-203, 1988

5. Bentley,J.H., "Multidimensional binary trees in database applications," IEEE Trans. Software Engineering, Vol.SE-5, No.4, pp.333-340, 1979

6. Brock,W.A., W.D.,Dechert and J.A.Scheinkman, "A test for independence based on the correlation dimension," Working Paper, Department of Economics, University of Wisconsin, Wisconsin, 1986

7. Broomhead,D.S., Private communication

8. Broomhead,D.S. and D.Lowe, "Multivariate functional interpolation and adaptive netwoks," Complex systems, Vol.2, pp.321-355, 1988

9. Dauito,B.J., T.T.Hartley and S.P.Chicatelli, *The hyperbolic map and application to the linear quadratic regulator*, Springer- Verlag, New York, 1989

10. Ding,H., C.Grebogi and E.Ott, "Dimension of strange nonchaotic attractor," Physics Letters A, Vol.137, pp.167-170, 1989

11. Eckhardt,B., "Irregular scattering," Physica D, Vol.33, pp.89-98, 1988

12. Eckmann,J.P., S.O.Kamhorst, D.Ruelle and S.Ciliberto, "Liapunov exponents from time series," Physics Review, Vol.34A, pp.4971-4979, 1986

13. Falconer,K., *Fractal geometry - mathematical formulation and application*, John Wiley and Sons, New York, 1990

14. Farlow,S.J.. *Self-organizing methods in modeling*, NY : Marcel Dekker, 1984

15. Franaszek,M., "Optimized algorithm for the calculation of correlation dimension," Physical Review A, Vol.39, No.10, pp.5440-5443, 1989

16. Goldstein,H., "Sea echo in propagation of short radio waves," MIT Radiation Lab. Ser., Vol.13, Sect 6.6, edited by D.E.Kerr, McGraw-Hill, New York, 1951

17. Grassberger,P. and I.Procaccia, "Measuring the strangeness of strange attractor," Physica 9D, pp.189-208, 1983

18. Grassberger,P., "Finite sample corrections to entropy and dimension estimates," Physics Letters A, Vol.128, pp.369-373, 1988

19. Greenside,H.S., A.Wolf, J.Swift and T.Pignarato, "The impracticality of a box-counting algorithm for calculating the dimensionality of strange attractor," Physical Review, Vol.A25, pp.3453-3456, 1982

20. Guckenheimer,J. and P.J.Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Springer-Verlag: New York, 1983

21. Haykin,S., *Adaptive filter theory*, Prentice-Hall, 1986

22. Hollowway,G. and B.J.West, *Predictability of fluid motion*, American Institute of Physics, New York, 1984

23. Hornik,K., M.Stinchcombe and H.White, "Multilayer feedforward networks are universal approximator," Neural Networks, Vol.2, pp.359-366, 1989

24. Jakeman,E. and P.N.Pusey, "A model for non-Rayleigh sea echo," IEEE Trans. Antennas and Propagation, Vol.AP-24, No.6, 1976

25. Lippman,R.P., "An introduction to computing with neural nets," IEEE ASSP Magazine, Vol.4, No.2, pp.4-22, 1987

26. Lorenz,E.N.,"Deterministic nonperiodic flow," Journal of Atmospheric Science, Vol.20, pp.130-141, 1963

27. Lorentz,G.G., *Approximation of functions*, Holt, Rinehart and Winston, New York, 1966

28. Mayer-Kress,G., "Application of dimension algorithms to experimental chaos," in *Directions in chaos*, edited by Hao Bai-Lin, World Scientific, Singapore, 1987

29. Milnor,J., "On the concept of attractor," Communication on Mathematical Physics, Vol.99, pp.177-195, 1985

30. Moody,I. and C.I.Darken, "Fast learning in networks of locally-tuned processing units," Neural Computation, Vol.1, pp.281-294, 1989

31. Passamante,P., T.Hediger and H.Gollub, "Fractal dimension and local instrinsic dimension," Physical Review A, Vol.39, No.7, pp.3640-3645, 1989

32. Parker,T.S. and L.O.Chua, *Practical numerical algorithms for chaotic systems*, Springer Verlag, New York, 1989

33. Saleh,B.E.A., "Optical bilinear transformation : general properties," Optica Acta, Vol.20, No.6, pp.777-799, 1979

34. Senjnowski,T.J. and C.M.Rosenberg, "Parallel networks that learn to pronounce English text," Complex systems, Vol.1, pp.145-168, 1987

35. Simmons,G.F., *Introdction to Topology and Modern Analysis*, McGraw-Hill, New York, 1963

36. Skolnik,M.I., *Introduction to radar systems*, McGraw-Hill, New York, 1980

37. Takens,F., "Detecting strange attractor in turbulence," in *Dynamical systems and turbulence, Warwick 1980* edited by D.A.Rand and L.S.Young, Lecture Notes in Mathematics, Vol.898, pp.366-381, Springer Verlag, 1981

38. Theiler,J., "Efficient algorithm for estimating the correlation dimension for a set of discrete point," Physical Review A, Vol.36, pp.4456-4460, 1987

39. Walsh,J.L., "The existence of rational function of best approximation," Trans. American Mathematical Society, Vol.33, pp.668-689, 1931

40. Widrow,B. and S.Stearns, *Adaptive signal processing*, Prentice-Hall, 1985

41. Wolf,A., J.B.Swift, H.L.Swinney and J.A.Vastano, "Determining Lyapunov exponents from a time series," Physica 16D, pp.285-317, 1985