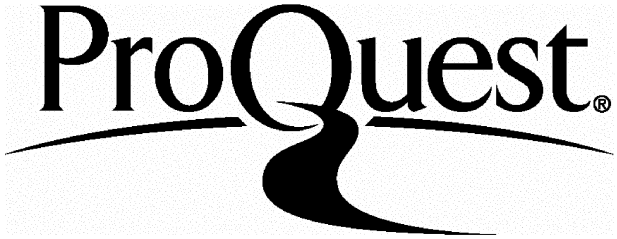
Noise Reduction in Nonlinear Time Series Analysis

Mike Davies

Centre for Nonlinear Dynamics
University College London

September 1993

Thesis submitted for the degree
of Doctor of Philosophy



Dedicated to the memory of my mother

Enid Davies

Abstract

Over the last decade a variety of new techniques for the treatment of chaotic time series has been developed. Initially these concentrated on the characterisation of chaotic time series but attention soon focused on the possibility of predicting the short term behaviour of such time series and we begin by reviewing the work in this area. This in turn has lead to a growing interest in more sophisticated signal processing tools based on dynamical systems theory. In this thesis we concentrate on the problem of removing low amplitude noise from an underlying deterministic signal. Recently there has been a number of algorithms proposed to tackle this problem. Originally these assumed that the dynamics were known *a priori*. One problem with such algorithms is that they appear to be unstable in the presence of homoclinic tangencies.

We review the original work on noise reduction and show that the problem can be viewed as a root-finding problem. This allows us to construct an upper bound on the condition number for the relevant Jacobian matrix, in the presence of homoclinic tangencies. Alternatively the problem can be viewed as a minimisation task. In this case simple algorithm such as a gradient descent algorithm or a Levenberg-Marquardt algorithm can be using to efficiently reduce noise. Furthermore these do not necessarily become unstable in the presence of tangencies. The minimisation approach also allows us to compare a variety of *ad hoc* methods that have recently been proposed to reduce noise. Many of these can be shown to be equivalent to the gradient descent algorithm.

The problem can then be extended to include the unknown parameters used to estimate the mapping function when the dynamics are unknown. Here we incorporate these into the noise reduction process and we show that this appears to improve the stability of the noise reduction algorithm. Finally we conduct a number of numerical investigations using our noise reduction algorithm. These include applications to data from the Lorenz equations and some experimental laser data.

ACKNOWLEDGEMENTS

First and foremost I would like to thank my parents for their encouragement over the years. I would also like to thank all the members of the Centre for Nonlinear Dynamics and my brother, Chris, for many interesting discussions.

CONTENTS

Introduction

Our aim in this thesis is to develop a theory for methods of reducing low amplitude noise from an underlying deterministic time series. For our purposes we can define a time series as a sequence of observations that are a function of time. Furthermore we only consider discrete time series and whenever examining continuous time series we will convert it into a discrete form by sampling it at a finite rate.

Traditional time series analysis is not new and many signal processing tools, including noise reduction techniques, have been developed based on such analysis. However traditional methods (FFT, ARMA, etc...) are, in general, restricted to linear transformations. This means that noise reduction techniques often involve identifying regions of the frequency domain where the desired signal predominantly lies. However the performance of such spectral methods is inevitably restricted if the noise is broad band.

A better approach would be to try to identify a discriminator that completely distinguishes between the wanted signal and the noise. Here we develop such an approach using low dimensional determinism to separate the two signals. This would not make a particularly interesting discriminator if we limited ourselves to traditional linear models since the resultant time series would merely have a number of discrete spikes in its frequency spectrum. However the same is not true for nonlinear models and possibly one of the most important lessons to be drawn from the study of nonlinear dynamics is that an apparently random signal, from a traditional statistical viewpoint, can be generated by a simple nonlinear deterministic system. Such signals are termed 'chaotic' and generally have broad band frequency spectra. Thus standard spectral methods for noise reduction are not directly applicable. This, and the recent growth in the study of experimental chaotic systems have led to a demand for good noise reduction techniques for such systems based on dynamical systems theory. It is this problem that we aim to address here.

To this end we begin in chapters 1 and 2 by reviewing state space methods for low dimensional dynamical systems and how to reconstruct an equivalent state space from a

single scalar time series. For this purpose we include a discussion on the important result of Takens' embedding theorem which demonstrates that, under suitable constraints, such equivalent state spaces exists and can be used approximate the dynamics and therefore to predict the short term furture behaviour of the time series.

We can then use these methods to construct various procedures of noise reduction. Initially we describe the noise reduction methods that are based loosely on the shadowing lemma. The shadowing problem can then interpreted as a rank deficient root-finding problem. In this context a noise reduction algorithm that has been proposed by Hammel can be seen as applying a set of reasonable constraints to the problem to make it full rank. However we demonstrate that even with these additional constraints the problem becomes ill-conditioned when the embedded trajectory becomes close to a homoclinic tangency. We then show that the problem can be reformulated as a minimisation task. This allows us to treat tangencies in a more stable manner by using either a gradient descent or a Levenberg-Marquardt minimisation procedure. It also provides us with a framework in which we can compare various other noise reduction schemes that have been proposed.

In chapter 4 we go on to extend these ideas to include the problem of modelling the dynamics within the noise reduction algorithm. This makes sense since both procedures aim to minimise the same approximation errors. We compare this approach to modelling the dynamics and reducing the noise separately. Our results show that intergrating the two steps together makes the noise reduction both more stable and more accurate.

In chapters 5 we investigate the performance of our methods under certain conditions. Finally, in chapter 6 we apply our noise reduction algorithm to time series from a variety of systems. These include both chaotic and non-chaotic attractors from the Lorenz equations and some experimental laser data.

1. Preliminaries

This chapter contains some of the basic definitions and concepts that are required in studying nonlinear dynamical systems. We have made no attempt to make this section complete and we have concentrated on the ideas that are specifically used in subsequent chapters. In section 1.1 we introduce definitions for continuous and discrete time dynamical systems. Then we discuss the possible solutions for such systems along with their stability properties. Here we include a definition of the important class of hyperbolic sets. Finally we define the Lyapunov exponents and explain how they gives us information about the stability of a solution. We also discuss a method for calculating these exponents for a given orbit since we will need to use this in chapter 3.

1.1 State Space Models and Dynamical Systems

We define a dynamical system as a map or vector field on some finite dimensional manifold M (usually compact) such that *either* (continuous time):

$$\frac{dx(t)}{dt} = f(x(t)) \ , \ t \in \mathbb{R} \qquad \textbf{(1.1.1)}$$

where $x(t)$ is a point on the d dimensional manifold M, *or* (discrete time):

$$x_n = f(x_{n-1}) \ , \ n \in \mathbb{N} \qquad \textbf{(1.1.2)}$$

where, again $x_n \in M$. We will also require in both cases that the dynamical system, f, is to some extent smooth (i.e. to have continuous derivatives to some order). For a discussion of the existence and uniqueness of solutions for such systems see, for example, Guckenheimer and Holmes [1983]. The manifold M on which the dynamics acts is then called the state space (it is also referred to in some texts as the phase space) and knowledge of the position of a point in state space defines a unique solution with respect to the dynamics.

In fact these ideas can be generalised to infinite dimensional dynamical systems, as long as the asymptotic dynamics can be restricted to some finite dimensional compact

manifold. This concept has been greatly enhanced in recent years by the work on the existence of inertial manifolds (see Constantin et al [1989] or Temam [1988]).

1.2 Stability of Solutions

We can now consider the solutions for such systems and their stability. We are particularly interested in some concept of 'steady state' solutions. For example, one simple steady state solution of equation 1.1.2 is a *periodic point*, which is a point, x, where there exists an n such that $f^n(x) = x$. The stability of a periodic point can then be evaluated from the eigenvalues of the derivative of the function at this point: Tf_x. This can be divided into stable and unstable subspaces. Solutions on the linear subspace, E_x^s, spanned by the stable eigenvalues, whose moduli $\|\lambda\| < 1$, will contract onto the fixed point at an exponential rate. Similarly points on the subspace, E_x^u, spanned by the unstable eigenvalues ($\|\lambda\| > 1$) will diverge away from the fixed point at an exponential rate.

However one of the interesting features of nonlinear dynamics is that complex *aperiodic* steady state motion is possible and we would like to be able to extend the idea of stability to more general invariant sets. An important concept in this respect is that of *hyperbolicity*. We take the following definition from Shub [1986]:

Definition: We say that Λ is a *hyperbolic set* for a map $f:M \to M$ if there is a continuous splitting of the tangent bundle of M restricted to Λ, TM_Λ, which is Tf invariant:

$$TM_\Lambda = E^s \oplus E^u; \quad Tf(E^s) = E^s; \quad Tf(E^u) = E^u \tag{1.2.1}$$

and there are constants $c > 0$ and $0 < \lambda < 1$, such that:

$$\begin{aligned} \|Tf^n|_{E^s}\| &< c\lambda^n \ , \ n \geq 0 \\ \|Tf^{-n}|_{E^u}\| &< c\lambda^n \ , \ n \geq 0 \end{aligned} \tag{1.2.2}$$

Here the linear subspaces E^s and E^u can be considered to be generalisations of the stable and unstable eigenspaces of Tf_x for the periodic case.

Finally we can extend these stability concepts from the infinitesimal to the local. Here we define the local stable and unstable manifolds in the ϵ neighbourhood of a point, x, in a hyperbolic set as follows:

$$W^s_\epsilon(x,f) = \{y \in M \mid d(f^n(x),f^n(y)) \to 0 \text{ as } n \to +\infty \text{ and } d(f^n(x),f^n(y)) \leq \epsilon, \forall n \geq 0\} \quad \textbf{(1.2.3)}$$

$$W^u_\epsilon(x,f) = \{y \in M \mid d(f^n(x),f^n(y)) \to 0 \text{ as } n \to -\infty \text{ and } d(f^n(x),f^n(y)) \leq \epsilon, \forall n \leq 0\} \quad \textbf{(1.2.4)}$$

These can be related to the linear subspaces discussed above by the stable manifold theorem (see Shub [1986] for a definition and proof) which states that these local manifolds have the same dimension as, and are tangential to their linear counterparts E^s and E^u. This idea is illustrated in figure 1.2.1.

Thus, if a set is hyperbolic, we know that in the neighbourhood of this set the solutions to the associated nonlinear map behave in a qualitatively similar way to the solutions in the linearised system. This is important when considering the properties of solutions close to a hyperbolic set and we will specifically use these ideas when discussing the *shadowing* problem in chapter 3.

1.3 Lyapunov Exponents

A hyperbolic set provides us with a splitting of the tangent space at a point into stable and unstable directions. We can now discuss the possibility of quantifying the expansion and contraction rates associated with these directions. Furthermore we are particularly interested in the existence of average expansion rates in the case of *aperiodic* motion. For this purpose we can take the following definition for Lyapunov exponents from Guckenheimer and Holmes [1983]:

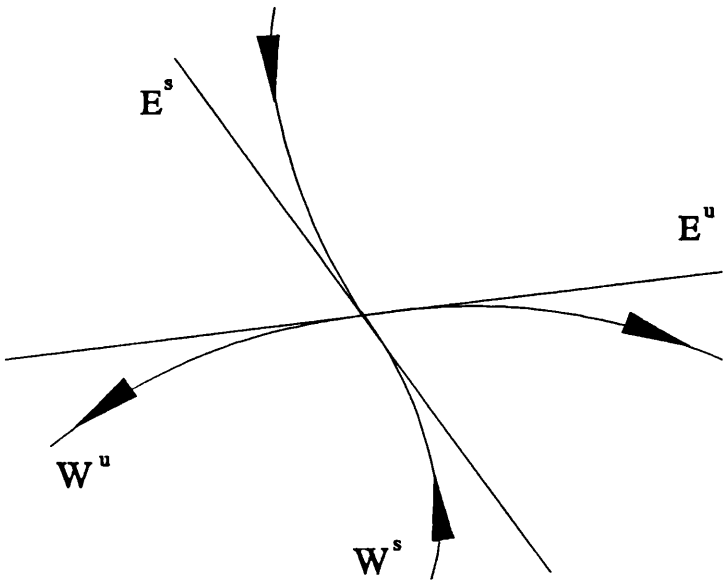


Figure 1.2.1: The invariant manifolds for a fixed point of a two dimensional mapping function f(x) and the linearised stable and unstable eigenspaces.

Definition: Let $f:M \to M$ and $x \in M$. Suppose there exist nested subspaces in the tangent space Tf_i at $f^i(x)$ such that $\mathbb{R}^n = E_i^{(1)} \supset E_i^{(2)} \supset \ldots \supset E_i^{(n)}$ and the following holds:

(1) $Tf(E_i^{(j)}) = E_{i+1}^{(j)}$

(2) $\dim E_i^{(j)} = n + 1 - j$

(3) $lim_{N\to\infty}\ ln\ (1/N) \| [(Tf^N)^T(Tf^N)]^{1/2}.v \| = \lambda_j,\ \forall\ v \in E_0^{(j)} \backslash E_0^{(j+1)}$

Then the set λ_j are the Lyapunov exponents of f at x.

It is clear from this definition that if x is a periodic point then there is a strong relationship between the Lyapunov exponents and the eigenvalues of the tangent map Tf_x. Indeed the Lyapunov exponents are simply the logarithm of the moduli of the eigenvalues. However the importance of the Lyapunov exponents stems from the fact that they are also applicable to aperiodic solutions such as chaotic attractors.

The concept of Lyapunov exponents can be extended to general invariant sets if we assume that the set possesses an invariant probability measure (i.e. one that is invariant under the action of f) and that this probability measure is ergodic (essentially that time averages equal spatial averages). We can then make use of the remarkable theorem of Oseledec [1968]:

Theorem (Eckmann and Ruelle [1985]): Let ρ be a probability measure on a space M, and $f:M \to M$ a measure preserving map such that ρ is ergodic. Let $T:M \to$ the $m \times m$ matrices be a measurable map such that

$$\int \rho(dx) \log^+ \| T(x) \| < \infty \qquad \textbf{(1.3.1)}$$

where $\log^+ u = max(0, \log u)$. Define the matrix $T^n = T(f^{n-1}x)\ldots T(fx)T(x)$. Then, for ρ-almost all x, the following limit exists:

$$\lim_{n\to\infty} \left((T_x^n)^T(T_x^n)\right)^{1/2n} = \Lambda_x \qquad \textbf{(1.3.2)}$$

Note that the logarithms of the eigenvalues of Λ_x are Lyapunov exponents, as defined above. This theorem then tells us that the Lyapunov exponents are constant almost

everywhere with respect to the invariant measure. A more detailed discussion of this theorem and a proof are given in Johnson, Palmer and Sell [1989].

Physically we can interpret the Lyapunov exponents as a measure of the sensitive dependence upon initial conditions. That is, the rate of exponential growth of an infinitesimal vector is given, in general, by the largest Lyapunov exponent. Similarly the exponential expansion of an infinitesimal k-dimensional surface is given by the sum of the largest k Lyapunov exponents. Note that this means that, if the system is dissipative then the sum of the Lyapunov exponents must be negative.

1.3.1 Calculating Lyapunov Exponents

Finally we consider how we could calculate the Lyapunov exponents for an attractor, given a fiducial trajectory from the system. Here we make use of the physical interpretation described above. That is we know that a vector chosen in the tangent space will, in general, align itself with the most expanding direction under the action of Tf^n. It will then asymptotically expand at the rate defined by the largest Lyapunov exponent. Similarly an k-dimensional linear subspace chose in the tangent space, under the action of Tf^n, will align itself with the k-dimensional subspace associated with the k largest Lyapunov exponents.

Hence we need to identify such nested subspaces from a fiducial trajectory. One method for doing this is to decompose Tf^n in terms of Tf_i (the derivative of f at the ith point in the trajectory) in the following way (see Eckmann and Ruelle [1985]):

$$\begin{aligned} Tf_1 Q_0 &= Q_1 R_1, \\ Tf_2 Q_1 &= Q_2 R_2, \\ &\cdots \\ Tf_n Q_{n-1} &= Q_n R_n. \end{aligned} \tag{1.3.3}$$

where Q_i is an orthonormal matrix and R_i is upper triangular with positive diagonal elements. Then we can write Tf^n as:

$$Tf^n\ Q_0 = Q_n \prod_{i=n}^{1} R_i \qquad \textbf{(1.3.4)}$$

Thus if we choose Q_0 to be the identity matrix this method provides a stable method for decomposing Tf^n into an orthonormal matrix and an upper triangular matrix (the product of upper triangular matrices is upper triangular). Thus the kth diagonal element of the product of upper triangular matrices provides information about the expansion of the direction in the k-dimensional linear subspace spanned by the first k vectors in Q_0 that is orthogonal to the k-1 dimensional subspace spanned by the first k-1 vectors in Q_0. Finally, since the diagonal elements of the product of upper triangular matrices are merely the product of the individual diagonal elements we can obtain an estimate for the kth Lyapunov exponent as:

$$\lambda_k = \frac{1}{N} \sum_{i=1}^{N} \ln R_i(k,k) \qquad \textbf{(1.3.5)}$$

where $R_i(k,k)$ is the kth diagonal element of R_i.

2. State Space Reconstruction

So far we have considered the modelling of dynamical systems in a state space. However in an experimental environment it may not be possible to measure all the variables to produce a state space. Indeed it may not be obvious what the actual state space is. It is therefore useful to consider how much information from a system is required to allow an adequate model to be constructed. This chapter reviews the work that has been done in this area with respect to modelling dynamics from one, or a few, scalar time series. The most important result here is Takens' embedding theorem which provides the basis for all the other methods discussed in this chapter. However before considering the theory in detail the motivation for the problem is set out.

2.1 Motivation

We have already mentioned that finite dimensional dynamical systems can be studied by constructing their state spaces, such that each point in the state space can be uniquely identified with a possible state of the physical system. This, and the knowledge of the dynamics, provides us with the ability to predict the future state of the system due to the uniqueness of the solutions for any given state. Our aim is to achieve something similar when faced with a time series measured from some dynamical system.

This is essentially the idea behind state space reconstruction. We are trying to construct a state space that is observable from the time series. This space therefore must reproduce the original state space under a smooth transformation, in some way. Then, if we can associate a point in the time series to a uniquely point in our reconstructed space, the dynamics can be realised by a map that takes these states to the next associated state in time.

In general the reconstructed space will be an n dimensional Euclidean space but the original state space on which the dynamics is defined may be some complicated m dimensional manifold. Thus the original state space will map into a submanifold of the

reconstructed space. Takens' embedding theorem demonstrates that it is possible to construct such a state space, thus opening up a whole host of opportunities for identification and prediction of time series from experimental dynamical systems.

2.2 Takens' embedding theorem.

Takens' observed that the effect of the dynamics on the time series provides us with enough information with which to reconstruct a space that contains a submanifold equivalent to the manifold on which the original dynamics acted. The proposed space is constructed using "delay coordinates", which are defined by a d-dimensional vector of the form:

$$\{y(x),\ y(\phi(x)),\ \ldots,\ y(\phi^{d-1}(x))\}$$

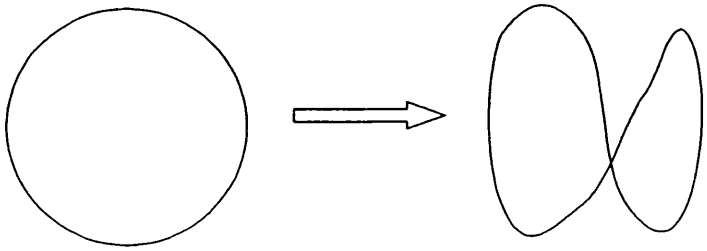
where $x \in M$ is a point in the manifold M on which the mapping function $\phi(x) : M \to M$ acts and y is the real valued function by which the dynamical system has been observed. If the dynamical system under consideration is continuous in time we can regard $\phi(x)$ to be the time τ map such that $x(t+\tau) = \phi(x(t))$.

Loosely speaking Takens' theorem shows that 'typically', as long as the dimension of the delay space is large enough, we have created a mapping from the manifold M into the delay space that is an embedding, where an embedding is defined as follows:

Definition: Let $f:M \to N$ be a smooth map. We say that f is an *immersion* if the tangent map Df is one-to-one everywhere. We say that f is an *embedding* if it is an immersion and everywhere one-to-one.

The concepts of immersions and embeddings are illustrated in figure 2.1.1 mapping $S^1 \to \mathbb{R}^2$. The top picture shows a mapping of a circle onto the plane that is an immersion, but not one-to-one. whereas the bottom picture shows a mapping that is one-to-one but not an immersion.

a.



b.

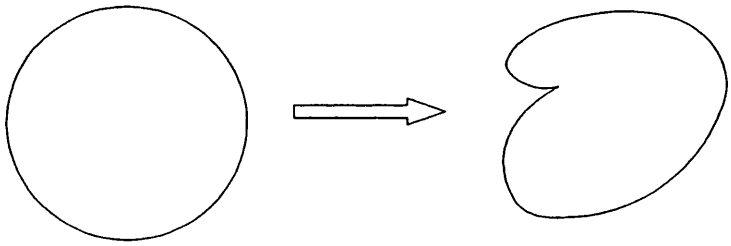


Figure 2.2.1: The top picture illustrates a mapping from a circle to the plane that is an immersion but is not one-to-one. The bottom picuture shows a similar mapping that is one-to-one but is not an immersion.

Finally the term 'typical' here means that this property is generic in the space of mapping functions and observables. Here we will use this to mean that the subset for which the property holds forms an open set that is dense within the relevant space (conversely this means the set where the property is not satisfied is closed and nowhere dense).

The strict statement of the theorem is as follows:

(Takens 1981) Let M be a compact manifold of dimension m. For pairs (ϕ,y), $\phi: M \to M$ a smooth diffeomorphism and $y:M \to \mathbf{R}$ a smooth function. It is a generic property that the map $\Phi(\phi,y):M \to \mathbf{R}^{2m+1}$ defined by:

$$\Phi(\phi,y)(x) = (y(x),\ y(\phi(x)),\ \ldots,\ y(\phi^{2m}(x))) \qquad x \in M \qquad \textbf{(2.2.2)}$$

is an embedding (smooth is at least C^2).

Sketch of Proof.

The proof of Takens' theorem follows closely to that of Whitney's embedding theorem (see Bröcker and Janich [1982] or Hirsch [1976]). However some additional work is required since the Φ is not a typical mapping function. We briefly discuss the method of the proof.

Initially we impose some restrictions upon ϕ. We assume that there is a finite number of points $x \in P_{2m}$, the set of periodic points in M with period less than $2m+1$ and that the eigenvalues of $D\phi_x^k$ are all distinct for $x \in P_{2m}$ of period k. These restrictions are acceptable since they are generic properties of ϕ. We further assume that y maps all the points in P_{2m} onto distinct points in $\mathbf{R}$. Finally we now note that both immersions and embeddings are open in the set of all mappings. Hence we only need to show that the set of (ϕ,y) for which Φ is an embedding is dense.

We begin by considering the set P_{2m}. It can be shown that, given, a perturbation of y

the derivatives dy_x, $dy\phi_x$,..,$dy\phi_x^{2m}$ span the tangent space TM_x for $x \in P_{2m}$. A detailed description of how to do this is given in Noakes [1991]. Thus for the map Φ associated with the adjusted y there is a neighbourhood V of P_{2m} such that $\Phi|_V$ is an immersion and locally an embedding.

We can now consider the points contained in $M \setminus V$. First we construct a finite open cover on $M \setminus V$, $\{U_i\}$ and a compact cover $\{K_i\}$ such that $K_i \subset U_i$ (some extra conditions also have to be imposed on $\{U_i\}$: see Takens' [1981]). We can then show that y can be perturbed to ensure $\Phi|_U$ restricted to some U is an immersion on a closed set $K \subset U$. This can be done by following the immersion proof in Bröcker and Janich [1982]. Hence for any U_i we can adjust y to immerse K_i. To make this property global we can then apply it individually to each open set, U_i, in turn composed with a suitable bump function whose support is contained in U_i. Since the set of immersions is open we can always choose the adjustment to make $\Phi|_{Ui}$ an immersion on K_i so small that our previous work is not destroyed.

Once Φ has been made an immersion on M, we can produce a new open cover $\{U_i\}$ such that each U_i is so small that $\Phi|_{Ui}$ is an embedding. Thus we only need to show that we can perturb y so that $\Phi|_{Ui} \cap \Phi|_{Uj} = \varnothing$ for $i \neq j$. Again we can follow the embedding theorem in Bröcker and Janich [1982] and the global results can be constructed in the same manner as above. This means that we can perturb y so that Φ it one-to-one on M as well as being an immersion. Hence Φ for the perturbed y is an embedding.

Recently these ideas have been extended by Sauer et al [1991] to investigate embeddings from a probabilistic point of view (generic does not necessarily guarantee probability one and some counter examples are given in Hunt et al [1992]). They also relaxed the smoothness condition from C^2 to C^1. Noakes [1991] has also produced an alternative proof for Takens' theorem which relaxes the smoothness condition to C^1. This can be done since every C^1 manifold has a compatible C^2 structure (see Hirsch [1976]).

2.3 Linear Filters and Embedding

It is worth considering the extensions of Takens' embedding theorem beyond delay embeddings to more general reconstructed spaces. One obvious candidate for an embedding space is linearly filtered time series. Often, intentionally or otherwise, the experimentalist's data will have already been filtered when it is analyzed, therefore it is necessary to ask whether a reconstructed space can be made from this data. The two types of linear filters described here are called Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters. An FIR filter is one with the following structure:

$$y_i = \sum_{j=0}^{n-1} a_j x_{i-j} \qquad (2.3.1)$$

where a_j are the constants of the filter and y_i is the time series resulting from filtering x_i. An IIR filter has an additional relation to the past values of y_i:

$$y_i = \sum_{j=0}^{n-1} a_j x_{i-j} + \sum_{k=1}^{m} b_k y_{i-k} \qquad (2.3.2)$$

where b_k are additional constants for the IIR filter. There are also acausal filters where the value of y_i is determined from data in the future as well as the past, but they are not considered here.

It is important to distinguish between these two types of filters since an FIR filter has only a limited "memory" of past data whereas an IIR filter, because it is recursive in nature, has a response that typically decays exponentially with time. This, it will be shown can lead to problems when filtering chaotic data. However first we should address the problem of FIR filters.

2.3.1 FIR Filters

The main result that shows that a bank of FIR filters can be used to create a reconstructed space was presented in Broomhead et al [1992] (a similar result based on measure-theoretical concepts was given by Sauer et al [1992]).

Theorem: *Let V be a time series of measurements made on a dynamical system (ϕ,M), which satisfies the hypotheses of Takens' theorem. Then for triples $(\mathbf{a},\phi,v)$, where $\mathbf{a} = (a_0,a_1,..a_n)$ are the constants of an FIR filter, it is a generic property that the method of delays, which constructs, from the time series U, vectors of the form:*

$$(u_j,u_{j-1},\ldots,u_{j-l+1}) \quad \text{where } l \geq 2m + 1 \text{ and } u_j = \mathbf{a}.v_j$$

with finite n, gives an embedding of M.

The proof of this theorem follows closely to that of Takens' original theorem. However it is possible to interpret this reconstructed space as a linear projection from a delay space of dimension $l+n-1$ to a new space of dimension l. Thus it is necessary to consider whether this projection destroys either uniqueness or differentiability. Broomhead et al showed that generically these properties were preserved if $l \geq 2m+1$.

2.3.2 IIR Filters

Unfortunately it is not possible to produce a similar theorem for IIR filters. This is because IIR filters are recursive in nature and therefore possess their own dynamics. Farmer, Ott and Yorke [1983] studied a one dimensional linear system that was driven by an observable from the cat map:

$$y_i = \alpha y_{i-1} + z_i \tag{2.3.3}$$

where y_i is the output of the linear filter, α is the contraction rate of the linear system, and z_i is the observable from the cat map. They noted that the torus of the original cat map when viewed in the extended dynamical system (that is including the linear system) ceased to be smooth when $\|\alpha\| > \lambda_-$ (the smallest Lyapunov exponent of the cat map).

However, the Lyapunov exponents are averaged properties of an invariant set. Therefore any relationship between the contraction rate of an IIR filter and the smallest Lyapunov exponent of the dynamical system will only provide information about smoothness *almost everywhere*. To see this we can consider the smoothness of the simple linear IIR filter given above, in the neighbourhood of a fixed point (this can easily be extended to periodic points). We can expand this filter in the following way:

$$y_i = \sum_{j=0}^{\infty} \alpha^j \, h(x_{i-j}) \tag{2.3.4}$$

where $\| \alpha \| < 1$ and $h(x_i)$ is the observation of the state variable x_i. Then we can regard the filtering as part of the observation process. That is, instead of using the observation function in Takens' theorem $h(x)$, we observe the system via $g(x)$, $g: M \to \mathbb{R}$, where $y_i = g(x_i)$. Then we can consider whether $g(x)$ satisfies the requirements for Takens' theorem to hold. In particular we are interested in the smoothness of the function, $g(x)$.

Consider a fixed point, p, of the observed dynamical system $p = f(p)$. Then we can write the partial differential of $g(p)$ along the direction of the eigenvector v associated with the smallest eigenvalue, γ of Df_p:

$$\frac{\partial g(p)}{\partial x}\Big|_v = \sum_{j=0}^{\infty} \alpha^j \, \frac{\partial h(p)}{\partial x}\Big|_v \, Df^{-j}\Big|_v \tag{2.3.5}$$

but, by our choice of v, we know that $Df^{-j}|_v = 1/\gamma^j$. Thus for the partial derivative of $g(x)|_v$ to remain bounded at p requires $\| \alpha/\gamma \| < 1$. This means that the delay map based on $g(x)$ will not be immersive at p. Furthermore, although $g(x)$ may still be C^1 small perturbations to $g(x)$ will not improve the situation. This is because, by the definition of the IIR filter, we are constrained to a very particular set of observation functions. Thus if $\| \alpha/\gamma \| \geq 1$ there will be at least one point on the manifold that is not emmbedded. This may be more severe than results based on the smallest Lyapunov exponent and we should really consider the Lyapunov exponents of every invariant measure on the attractor to ensure that the reconstructed manifold is smooth.

2.3.3 Singular System Analysis

A more sophisticated use of linear FIR filters is to construct a set of linearly independent (with respect to the time series) vectors using singular system analysis (this is also known as Karhunen-Loève decomposition or principal component analysis). This was proposed by Broomhead and King [1987] for over-sampled data from deterministic flows and

effectively uses a bank of orthogonal FIR filters for the reconstructed space. It is based on the trajectory matrix formed from the data set using the method of delays. First the data must be organised into vectors or 'windows' of a certain length. Then the matrix takes the following form:

$$X = \begin{pmatrix} x_1 & x_2 & x_3 \ldots . x_m \\ x_2 & x_3 & x_4 \ldots . x_{m+1} \\ . & . & . \quad . \\ . & . & . \ldots . x_{m+n} \end{pmatrix} \qquad \textbf{(2.3.6)}$$

where m is the window length. The trajectory is thus made up of m-windows of the data. We can now construct basis vectors in the m-window that are linearly independent with respect to the time series. This is done by diagonalising the covariance matrix X^TX using eigen-decomposition:

$$XX^T = V\, S^2\, V^T \qquad \textbf{(2.3.7)}$$

Where S^2 is a diagonal matrix containing the eigenvalues of X^TX and V is the matrix whose columns are the eigenvectors associated with S^2. Since the covariance matrix is the product of X^T and X it can be shown that it is symmetric and positive semi-definite. Thus the eigenvalues are all positive and the eigenvectors are mutually orthogonal. We can now define m new time series that are linearly independent by rotating the trajectory matrix X with V to get XV. These new directions are the principal directions in which the data lies and the eigenvalues define the extent to which the data lies in the associated direction. These are also called the principal values. S and V can also be calculated from the singular value decomposition of the trajectory matrix, $X = USV^T$.

However there is no point in merely rotating the elementary delay space. Therefore we wish to choose a subset of a large delay space such that the data embeds in this subspace. The basic idea is to determine a linear subspace within the delay space that predominantly contains the data. Obviously, if the data comes from a nonlinear dynamical system, the data will in general not lie completely on a linear subspace of the delay space. However when the data is over-sampled from a deterministic flow (this was the application proposed by Broomhead and King [1987]) then the delay space can be divided into two

distinct sets by considering the effects of noise. In such a situation, it is likely that some of the principal values of the trajectory matrix will be small since, if the window length is chosen to be such that its time span $\tau_w \leq 2\pi/\omega$ where ω is the band limiting frequency, the principal values will drop off quickly. Then the data will lie predominantly in linear subspace associated with the first few principal values. This can be seen as the redundancy of over-sampling.

We can now divide the simple delay space into linear subspaces that do and do not predominantly contain the embedded data by considering the effect of any additive noise. If the signal is corrupted by white noise then the covariance matrix will take the following form:

$$\Xi = \bar{\Xi} + \langle \varepsilon^2 \rangle I \tag{2.3.8}$$

where I is the identity matrix and ϵ is the variance of the additive noise. The new principal values are simply modified by:

$$\sigma_i^2 = \bar{\sigma}_i^2 + \varepsilon^2 \tag{2.3.9}$$

The space can then be divided into two parts. The subspace with principal values that are very much greater than ϵ^2 and the subspace with singular values approximately equal to ϵ^2. We can now consider the data projected into the first principal directions to be approximately deterministic. These can be considered the best directions in the sense that they have maximum variance and hence maximum signal-to-noise ratio. However the number of dominant principal values has no relationship with the embedding dimension of the data. This is clear since raising the noise level will reduce the number of dominant principal values whereas reducing it will increase their number, independent of the true embedding dimension.

Although singular system analysis is fundamentally a linear approach if the data lies predominantly on a linear subspace we can hope that maximising the signal-to-noise from a linear perspective will provide a good workable state space. Gibson et al [1992] developed this idea and showed that if the window width in the analysis tends to zero then the eigenvectors tend to normalised Legendre polynomials. In turn these can be seen as being proportional to the derivatives of the time series (again when the window width

tends to zero).

To illustrate this form of embedding we took a 5000 point time series from the Lorenz equations, using a delay window of length 10. The differential equations are:

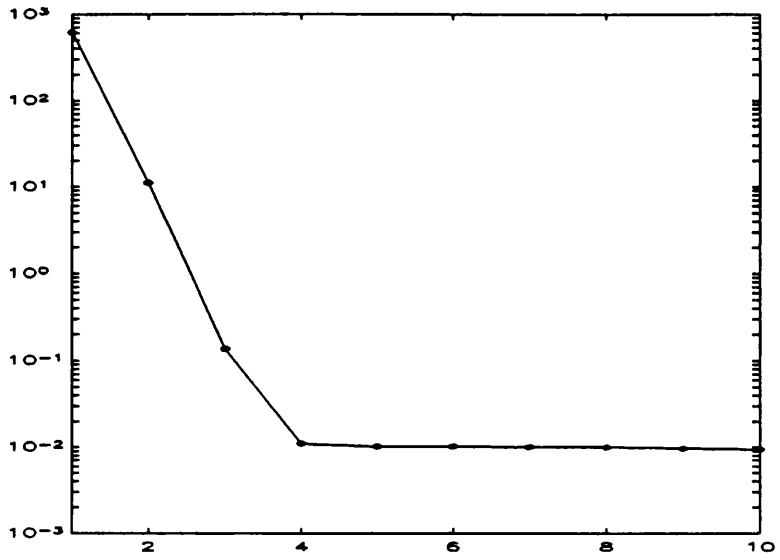
$$\begin{aligned} \frac{dX}{dt} &= \sigma\,(Y - X) \\ \frac{dY}{dt} &= rX - Y - XZ \\ \frac{dZ}{dt} &= -bZ + XY \end{aligned} \qquad \textbf{(2.3.10)}$$

where $\sigma = 10$, $b = 8/3$ and $r = 28$. The equations were integrated using a Runge-Kutta fixed step integration routine with a step size of 0.001. After the transients had died down a time series of 5000 points was obtained. The time series was then corrupted with 1% additive white noise. The top picture in figure 2.3.1 shows the plot of the normalised eigenvalues for the covariance matrix using a delay window of 10 points. It is clear from the figure that the singular values can easily be divided into the first three that are above the noise floor and the others.

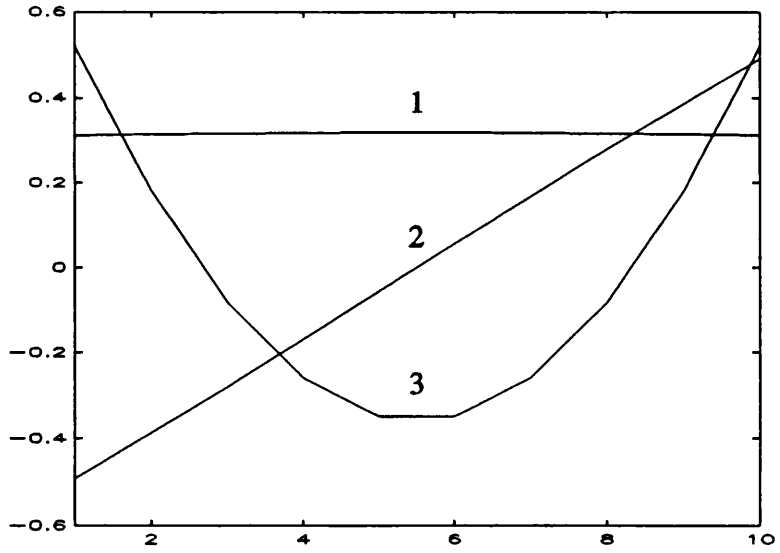
The first three associated principal directions are also shown in figure 2.3.1 and there is obviously a strong resemblance between these basis functions and the first three Legendre polynomials. The projections of the delay space into the associated principal directions are shown in figure 2.3.2.

2.4 System Identification

The theorems reviewed in the last sections demonstrate that it is possible to reconstruct a state space using some transformation applied to a delay embedding of a sufficiently high dimension. However, in practice, data will be corrupted by noise and will only have a finite length. This will, in general, result in some embeddings being better than others. This is a statistical concept whereas the embedding theorems are geometrically based and therefore they do not provide any direct insight into how to choose an optimal embedding



Singular Number



Delay Window

Figure 2.3.1: The top picture shows a plot of the singular values of the trajectory matrix for data from the Lorenz system. The bottom picture shows the shape of the first three associated singular directions.

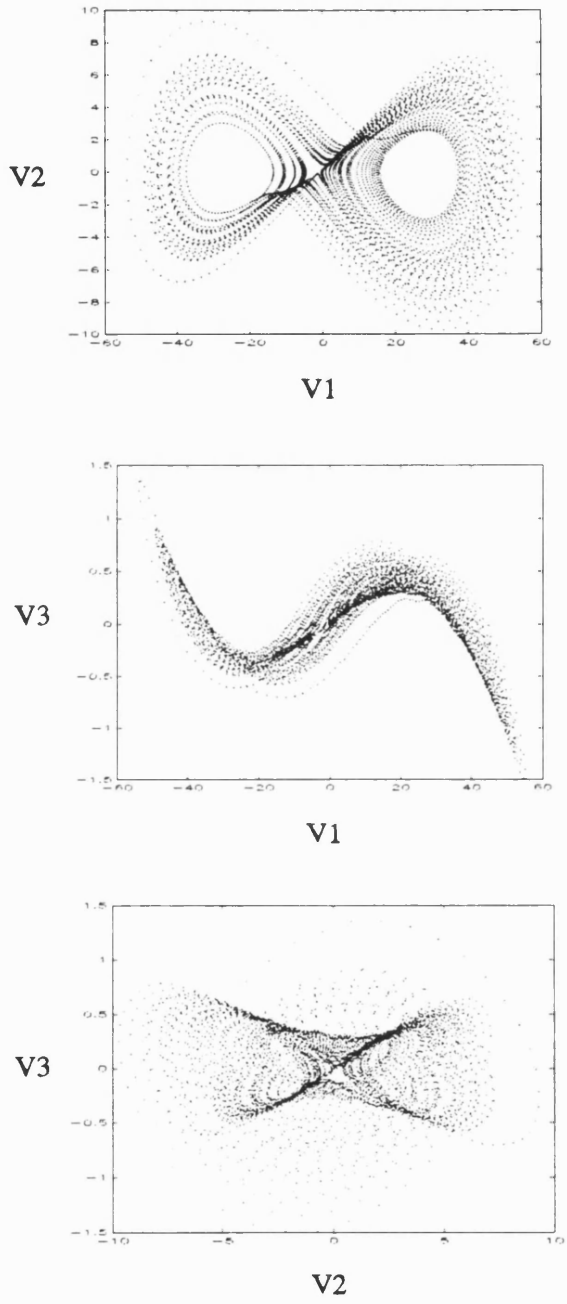


Figure 2.3.2: Three projections of data from the Lorenz system using the first three principal directions and a simple ten point delay window.

(i.e. what dimension and delay vectors to use).

It has previously been stated, [Kennel et al 1992], that the problem of choosing an embedding dimension and the problem of choosing delay time (the latter can be generalised to the choice of any linear filtered state space) are independent since the former is a geometric problem and the latter is a statistical problem, however this is not the case. Even when the ideal situation of no noise and infinite data length is considered it can be shown that delay times and dimension choice are linked. Although Takens' theorem states that the data will generically be embedded when the embedding dimension, d_E, is greater than $2m+1$ it is commonly known that many data sets can be embedded in lower dimensional spaces (Whitney showed that there always exists an embedding of an m-dimensional manifold in $2m$ dimensions: see, for example Sauer et al [1992]). Thus for many data sets there are open sets of the pairs $\{\phi,y\}$ for which the manifold containing the data is embedded in a reconstructed space for $d_E < 2m+1$. However a corollary to Takens' theorem is that for $d_E < 2m+1$ the space of the pairs $\{\phi,y\}$ will always contain open sets for which the manifold is not embedded and the dimension of the bad set in these cases is $2m-d_E$. These can easily be found by choosing an observation function such that two parts of the manifold are mapped into the same local neighbourhood in the state space then generically the intersection of the two local parts of the manifold will be $2m-d_E$. Therefore it may be possible to have an embedding for one choice of delay space for $d_E < 2m+1$ whereas another choice of delay space may require $d_E = 2m+1$ before an embedding is possible. The problem becomes even less well defined when finite noise levels are considered.

It is not obvious at present what the best approach to obtaining a good embedding is. Even so there has been much work done on constructing systematic ways of approaching these problems, some of which are described below. Casdagli et al [1991] provided a good review of the main criteria by which the dimension (and sometimes the delay time) of the reconstructed space can be optimised. The basic method that is used (independent of the criteria) is to repeatedly add new state vectors to the state space until some cost function ceases to increase with embedding dimension. The two broad categories of criteria that have been investigated are information theoretic measures and predictability measures. These are briefly discussed below.

2.4.1 Mutual Information

Shaw [1985] proposed using the concepts of *mutual information* to achieve an optimal embedding and this idea was taken up by Fraser and Swinney [1986] and Fraser [1989], where they argued that the choice of dimension and vectors used in singular systems analysis was intrinsically linear, whereas optimal choices of coordinates should be based on nonlinear criteria. It was proposed that minimising the mutual information between coordinates would provide coordinates that were most independent and therefore in some sense optimal. The mutual information between two variables x, and y is $I(x,y) = H(x) - H(x|y)$ where $H(x|y)$ is the entropy associated with the conditional probability of x given y. Thus it measures how much information y provides about the variable x. Unfortunately this idea is flawed for two main reasons.

First of all a chaotic attractor has positive entropy and therefore creates information. This means that as the delay time, τ, tends to infinity the mutual information between $x(t)$ and $x(t+\tau)$ tends to zero: this is obviously not optimal! To overcome this problem Fraser suggested choosing the first local minimum. However the mutual information function may not always have a minimum. In fact, it is quite common in maps for the mutual information function to monotonically decrease with τ since the positive entropy is the dominating factor.

The main problem of this method is the heuristic idea that maximum information will be gained about the state space by choosing a variable that is most independent from the previously chosen variables. This is not necessarily true and it would be a more justifiable approach to explicitly aim to maximise the information about the state (this was the idea originally proposed by Shaw).

Finally the success of using information theoretic methods in general is very dependent on what the reconstructed space is intended for. Casdagli et al [1991] pointed out that information theoretic measures do not impose any *a priori* constraints on the shape of probability distributions. In this case we are interested in minimising the uncertainty of the position in state space for a given point in the time series. An information theoretic measure will favour any probability distribution that is highly peaked whereas we may

intend to approximate the dynamics in the state space with a method that requires the variance of the probability distribution to be small (this is essentially the case when we aim to construct a one-to-one map). Therefore we favour the use of predictability criteria.

2.4.2 Predictability Criteria

In this section we discuss error estimates as criteria for estimating when the data has been successfully embedded. The main idea is that an error function provides a good measure of how well the embedding can be considered to possess a *one-to-one* mapping from the embedded data onto itself. This method has been implemented in a variety of different forms: Aleksic [1991], Savit and Green [1991], Cenys and Pyragas [1988], Kennel et al [1992] and Sugihara and May [1990]. Although these methods are similar their implementations and the interpretations given to the methodology vary significantly. Some of the interpretations are geometric in nature. Others are more statistical.

The main idea of the above papers is based on local techniques although there appears to be no reason why this cannot be extended to global descriptions of the embedding (see the end of this section). The method depends on the effect of increasing the embedding dimension on nearby points. If the data is fully embedded in some reconstructed space of dimension d then the addition of an extra state variable will create a new space in of dimension $d+1$ in which the data will also be embedded. Hence two close points will remain close in a higher dimensional space. This idea is illustrated pictorially in figure 2.4.1.

The geometric interpretation of this method can be made by choosing a distance threshold such that close points in dimension d whose distance is above the threshold when measured in a $d+1$ dimensional state space are said not to be embedded. When all the data remains close enough to near by points in the previous dimension then the data can be considered to have been successfully embedded to within this tolerance. Savit and Green [1991] noted that the embedding dimension defined in this way is inevitably a function of the tolerance chosen.

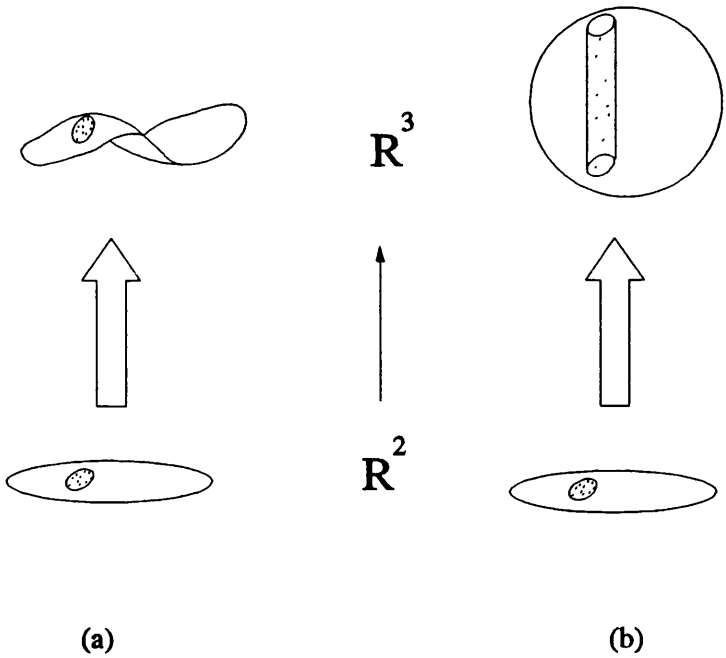


Figure 2.4.1: An illustration of the method of estimating the embedding dimension using nearby points. Both pictures show a two dimensional space being extended into three dimensions. In (a) the data lies on a two dimensioal surface and the points remain close. In (b) the embedding dimension is greater than two and the points are mapped far apart.

It should be noted that the closeness test may not itself be necessarily straight forward. For example the threshold criterion used in Kennel et al [1992] is:

$$\left[\frac{R_{d+1}^2 - R_d^2}{R_d^2}\right]^{\frac{1}{2}} > R_{tol} \qquad \textbf{(2.4.1)}$$

where R_d is the distance between a point and its nearest neighbour in the d-dimensional reconstructed space. However they noted that this criterion alone is not a satisfactory measure since, for finite data, as the dimension increases the distance between nearest neighbours tends to the span of the data set in that dimension, hence the criterion above will tend to zero as the dimension, d, becomes large. To solve this problem Kennel et al introduced a second test to determine whether the distance of nearest neighbours was too large to regard the data points as close.

We can see why this happens by considering the relation between R_d^2 and R_{d+1}^2:

$$R_{d+1}^2 = R_d^2 + R_1^2(x_{d+1}, y_{d+1}) \qquad \textbf{(2.4.2)}$$

where $R_1^2(x_{d+1}, y_{d+1})$ is the distance between x and y in the new coordinate direction. Thus it is obvious that when R_d^2 becomes very large the test in equation 2.4.1 will fail to be of use. However, if we only consider the size of $R_1^2(x_{d+1}, y_{d+1})$, we automatically get around this problem.

This then leads us to a statistical interpretation of the method since what we are measuring with $R_1^2(x_{d+1}, y_{d+1})$ is the mapping error associated with the local zeroth order map defined by the nearest neighbour (see section 2.5.3 for more details) which maps the d-dimensional reconstructed space into the new coordinate direction. Thus we are considering the data to be embedded when all the errors are sufficiently small. Once the concept is viewed in this light it is possible to consider more generalised error measures. For example, we could choose our optimal embedding dimension when the mean squared error or the absolute error is sufficiently small. Sugihara and May [1990] split the data into two halves and then used a local linear interpolated map constructed on the first half to predict the second half. On increasing the dimension the correlation coefficient (another measure of predictability) increased until it reached a plateau.

Finally there is no reason why the predictability criterion cannot be extended to global descriptions of the embedding. That is we could use the prediction error from a global function approximation for the dynamics (see section 2.5.2) as a measure to ascertain when the data has been embedded. Indeed, if our ultimate aim is to produce a predictive model for the data, it would seem that a good criterion for optimising our state space parameters would be the success of our predictive model.

2.4.3 Further Optimisation

The methods described above construct embeddings by repeatedly choosing an additional delay vector that is optimal in some sense with respect to the present set of state vectors, until the improvement is insignificant. However this does not necessarily optimise the whole space. To attempt to find such an optimal state we would need to search through the set of all reconstructions that are to be considered. In this respect, Meyer and Packard [1992] have looked at determining local embeddings for high dimensional attractors. They used a predictability criterion to choose the optimal local embedding for a given point in a high dimensional attractor. Obviously this problem is going to be nonlinear and to obtain a good minimum a sophisticated minimisation technique is required. Meyer and Packard [1992] solve this by using a genetic algorithm. However, in general, if a global embedding is required the gains are unlikely to warrant the extra effort.

2.5 Function Approximation

Once the data has been embedded into a reconstructed space it is possible to estimate the dynamics that produced the time series. This can be done by approximating the function $f:\mathbf{R}^d \rightarrow \mathbf{R}^d$ that maps the state space onto itself.

If the reconstructed space has been produced from a single scalar time series (this is the only case that will be considered here, although it is easy to extend the ideas to the more general case) then all that is required to approximate $f(\underline{x})$ is to model the mapping $g:\mathbf{R}^d \rightarrow \mathbf{R}$ from the reconstructed space onto the real line such that g predicts the next

$g:\mathbf{R}^d \rightarrow \mathbf{R}$ from the reconstructed space onto the real line such that g predicts the next point in the time series:

$$x_n \approx g(x_{n-1}, x_{n-2}, \ldots x_{n-d}) \qquad (2.5.1)$$

then the remainder of the new point in the state space is defined by a simple shift in the delay window so that the new state vector is:

$$\underline{x}_n = \{g(x_{n-1}, x_{n-2}, \ldots x_{n-d}), x_{n-1}, x_{n-2}, \ldots, x_{n-d+1}\} \qquad (2.5.2)$$

Once the dynamics have been modelled the system can be investigated further: e.g. prediction, control, filtering, etc...

2.5.1 Interpolation, Extrapolation and Approximation.

If the data is noise free and arbitrarily long then the dynamics will be uniquely defined on the embedded data set. However, in reality, a data set will always have a finite length. Therefore, in general we will not be able to ascertain a prediction directly from the data and it will be necessary to interpolate between data points. Also, in some cases, it may be useful, although more dangerous, to extrapolate from some data points. This requires us to select a subset of the relevant function space since there are an infinite number of functions that can be interpolated through any finite data set.

Furthermore, if noise is present in the data, then we will no longer wish to merely interpolate between the data points since this will wrongly incorporate the noise into the function model. In such a case we have to fall back on statistical concepts by approximating the data using a function model that is sufficiently constrained so that it does not fit the noise. The most commonly used method is to minimise some approximation error, although other methods, such as regularisation (for example see Press et al [1992]), do exist. Here we will restrict ourselves to minimising an approximation error defined by an ordinary least squared estimate. In the case of fitting a prediction function of the form given in equation 2.5.1 the error function can be written as:

$$H = \sum_{i=d}^{n} \left(x_i - g(x_{i-1}, x_{i-2}, \ldots, x_{i-d}) \right)^2 \qquad (2.5.3)$$

Although the least squared estimate is statistically based on quite strict assumptions about the nature of the noise present it does, in general, provide a good estimate for function approximation even when some of the assumptions are relaxed. This and the fact that it can often be solved directly makes it our preferred choice.

Finally we have to consider the choice of the function set from which we intend to make our approximation. Unfortunately it is difficult to identify what defines a good approximation function and most of the function forms used are, at best, based on weak ideas about the type of functions that are good and, at worst, on historical accident. Although there are no real optimal methods some techniques have advantages over others. There are two main categories of function approximation that have been applied to fitting nonlinear state space models. These are briefly discussed below.

2.5.2 Global Function Approximation

This approach restricts the approximation to a single set of functions that can usually be defined by a parameter set, p. Since there are no rigorous arguments for choosing one function form over another the speed at which a solution can be obtained plays a major role in the type of function forms that are used. In practice, this means that the function model should be chosen such that the parameters can be calculated directly which requires the model to be linear with respect to the parameters:

$$g(\underline{x}) = \sum_{i=1}^{p} w_i \, \phi_i(\underline{x}) \qquad (2.5.4)$$

where $\phi_i(\underline{x})$ is the ith predefined basis function and w_i is its associated weight. It is important to note that, as long as the basis functions are nonlinear there is no problem with the approximation being linear in parameter. Indeed it is a positive advantage since it allows us to calculate the weights for a given data set directly.

We can see this by considering the error function in terms of the weighting parameters:

$$H = \sum_{i=d}^{n} \left(x_{i+1} - \sum_{j=1}^{p} w_j \, \phi_j(\underline{x}_i) \right)^2 \qquad (2.5.5)$$

Because the function model is linear in parameters the error function is quadratic in parameter and has a unique minimum that can be calculated directly using linear algebra. Equation 2.5.5 can be rewritten in matrix form:

$$H = \boldsymbol{e}^T\boldsymbol{e} = (\boldsymbol{x} - \boldsymbol{P}\boldsymbol{w})^T(\boldsymbol{x} - \boldsymbol{P}\boldsymbol{w}) \qquad (2.5.6)$$

where e is the error vector for the time series, x is the vector of points from the time series to be predicted and P is the design matrix that has the following form:

$$P = \begin{pmatrix} \phi_1(\underline{x}_1) & \phi_2(\underline{x}_1) & \cdots & \phi_p(\underline{x}_1) \\ \phi_1(\underline{x}_2) & \phi_2(\underline{x}_2) & \cdots & \phi_p(\underline{x}_2) \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \phi_1(\underline{x}_{n-d}) & \phi_2(\underline{x}_{n-d} & \cdots & \phi_p(\underline{x}_{n-d}) \end{pmatrix} \qquad (2.5.7)$$

whose ij element is the value of the jth basis function at the ith state vector. We can now solve for the minimum value of H by differentiating and equating to zero:

$$\frac{\partial H}{\partial w} = 2P^T(x - Pw) = 0 \qquad (2.5.8)$$

Hence the values of w_j that minimise the error are:

$$w = (P^TP)^{-1}P^Tx \qquad (2.5.9)$$

At this point we should note that calculating the covariance matrix, P^TP, is not the only way in which to solve this least square problem. $(P^TP)^{-1}P^T$ is the Moore-Penrose pseudo-inverse for P and can be directly calculated for P using singular value decomposition (for example, see Golub and Van Loan [1983]). Similarly other orthogonalisation methods can be used (i.e. Gram-Schmidt). These methods also have the benefit of enabling the user to calculate truncated approximations to the least squared problem. This is particularly

useful when P is badly conditioned.

Thus we have shown that it is possible to reduce the approximation of nonlinear functions to a simple linear problem that can be solved directly by matrix inversion. However this method requires that the function model is linear with respect to the weighting parameters. We will now discuss below some of the function models that fit into this category:

Polynomials. Polynomial expansions are probably the most obvious choice of function form and can easily be written as a linear sum:

$$g(\underline{x}) = a_0 + a_1\underline{x} + a_2\underline{x}^2 + a_3\underline{x}^3 \ldots \qquad \textbf{(2.5.10)}$$

However when the order of the polynomial or the dimension of the domain become large the number of free parameters grows rapidly and will in general such fits are most useful when the order of the polynomial can be kept small (for example in local function approximation). A better alternative is to use rational polynomials.

Rational Polynomials. These are a simple extension from the polynomial expansion. A rational polynomial is the ratio of two polynomial expansions:

$$g(\underline{x}) = \frac{p}{(1 + q)} \qquad \textbf{(2.5.11)}$$

where both p and q are polynomial expansions. The above equation is obviously not linear in parameters. However direct solutions can be obtained by minimising a slightly different error function (see Casdagli [1989] and the references therein):

$$H^* = \sum_{i=d}^{n} \left((1 + q(\underline{x}_i))x_{i+1} - p(\underline{x}_i) \right)^2 \qquad \textbf{(2.5.12)}$$

This cost function can now be minimised directly and in the case of strict interpolation this solution will give be a zero solution (and a minimum) for the cost function H (equation 2.5.5). However this is unlikely to be the case when rational polynomials are being used for approximation and it is not at all clear exactly how this will effect the quality of the function approximation.

Radial Basis Functions. These provide an alternative global function fit that is not based on polynomial expansions. Radial basis functions are functions that are solely dependent upon some distance measure, usually Euclidean, from the basis function's centre, c. Thus the global model takes the following form:

$$f(\underline{x}) = \sum_{i=1}^{p} w_i \, \phi(\|\underline{x} - \underline{c}_i\|) \qquad \textbf{(2.5.13)}$$

where w_i is the weight associated with the ith basis function. It can be seen that, as long as the centres are defined *a priori*, this model is linear in parameters and can therefore be solved directly.

Radial basis functions were initially introduced as a means of performing multi-dimensional function interpolation (see, for example, Powell [1985]). In this case the centres are automatically chosen as the data points themselves. The basic incentive in using radial basis functions is that they can exhibit good localisation properties. That is the value of the resulting function at a point is mainly defined by nearby points in the space. In this sense radial basis functions are similar to the local methods described in the next section. They can also be shown to possess other desirable properties: see, for example Michelli [1986]. One example of a radial basis function model is the basis function $r^2 log\ r$. If this basis set is applied in 3 dimensions it can be shown to be equivalent to interpolating the data with thin plate splines. Some other commonly used basis functions are: $exp(kr)$, $(r^2 + k)^{1/2}$, $(r^2 + k)^{-1/2}$ and r^3. In each case k is an arbitrary scaling factor that has to be predefined.

Casdagli [1989] originally proposed using radial basis functions for interpolating between data from a time series in a reconstructed state space. However this ignores the problems of noise. This was addressed by Broomhead and Lowe [1988] who showed that if there were less basis functions than data points this method could be used for multi-dimensional function approximation. They also showed that such a function model could be viewed as a neural network with a specific architecture that allows the weights to be calculated directly.

Finally it is important to mention that there are methods of nonlinear function approximation that are not linear in parameter, such as neural networks. However these methods require iterative procedures to minimise the error function and therefore the fitting procedure takes orders of magnitude longer. For this reason we favour function approximations such as radial basis functions which appear to be able to achieve good fits without being computationally too costly.

2.5.3 Local Function Approximation

Another approach to function approximation is to construct local models. The incentive for this is that, if the surface to be approximated is quite complicated then a global approximations may perform poorly. However if we assume that the function is to some extent smooth we can exploit the local structure that this imposes. Consider the Taylor expansion of a function f around a point in state space x:

$$f(x + \delta x) \approx f(x) + \delta x \frac{df}{dx} + \delta x^T \frac{d^2 f}{dt^2} \delta x + \ldots \qquad \textbf{(2.5.14)}$$

Then we know that the function of a small enough neighbourhood of x is closely approximated by its Taylor expansion to some order. Thus, instead of approximating the function globally, we can approximate the Taylor expansion locally.

To implement this method we need to define a metric with which to construct a neighbourhood of the point we wish to predict. We then find all the points in the training set that lie in that neighbourhood and fit a local model over the neighbourhood in the same way that was described in the last section. The most natural function model to use when approximating a Taylor expansion is to use a simple polynomial expansion. Unlike global polynomial expansions, the order of the polynomial can be kept small such that each individual fit is well behaved.

The simplest model is the zeroth-order model where the value of the function over the neighbourhood is approximated by a local constant. Hence the global model is a piece-wise constant fit. This was originally used by Lorenz [1969] in an attempt to make

atmospheric forecasts. However the most popular method is to use local linear approximation (see, for example Farmer and Sidorowich [1987] and Sugihara and May [1990]). This also provides derivative information which is useful for calculating Lyapunov exponents and also for performing noise reduction (see chapter 3). Higher order polynomials have also been investigated by Farmer and Sidorowich [1988]. Although some improvement can be gained by increasing the order of the polynomial this gain is likely to be limited by the size and accuracy of the data set.

However we have so far not specified how to chose a neighbourhood. The most natural way is to chose all the data points that lie within a specific distance from the point, y, to be predicted. However it is not obvious what the distance should be or whether there will be enough points within the neighbourhood to make a good prediction. Hence another method for choosing a neighbourhood of y is to find the k nearest data points to the y. This then guarantees that there will be enough points to make the approximation well defined. Finally a more systematic approach that has been used to define neighbourhoods in a different context was presented by Broomhead et al [1987] in calculating topological dimension of attractors. They looked at the scaling of the local prediction error with the neighbourhood size, σ. When the scaling changes as σ^2 then the neighbourhood is becoming too big. Therefore it should be possible to devise an algorithm to use the maximum possible neighbourhood before curvature errors become dominant.

Finally, once a mechanism for choosing neighbourhoods has been found and the type of local function approximation has been chosen, we need a method for efficiently searching through the data set to locate the nearest neighbours. If we were to use a simple sequential search we would soon find that the this type of function approximation was too costly, since we would be spending all our time searching through the data set. However there are fast search algorithms available that speed up this process to an acceptable level. These generally involve partitioning the data in the state space and then ordering these regions in a manner that is easy to search. Two examples of quick search algorithms that have been used in local function approximation are k-d trees and box-assisted algorithms. See Bentley and Friedman [1979] for a full comparison of the various methods available.

One major problem with local function approximation is that it does not allow the

functional form to be described by a parameter space. Although parameters are calculated for the functions in each neighbourhood this creates a vast number of free parameters many of which are unlikely to be independent. Thus it is not easy to consider a family of functions related to perturbations of the data set. Although this appears to be a small point we will show in chapter 4 that such a family of functions can be used effectively to enhance the performance of noise reduction algorithms.

2.6 Approximation Errors versus Measurement Noise

In the next chapter we will discuss methods by which the noise in the time series can be reduced. However before looking at these techniques it is necessary to consider how different types of noise will effect the data in the reconstructed space. Broadly speaking, dynamical systems can be contaminated by two types of noise:

> **Dynamic noise** is a stochastic process within the dynamical system such that the actual trajectory is perturbed by the random process.
>
> **Measurement noise** is a random component that corrupts the time series but does not effect the dynamics or the underlying trajectory.

We will restrict ourselves to additive dynamic noise and additive measurement noise, although in practice, noise does not have to take these forms.

From the description that we gave of the function approximation methods it is clear that the resulting approximation errors act as additive dynamic noise. This is irrespective of the actual type of noise present. However, if we believe that our time series has been corrupted by some additive measurement signal (for example we may have some prior knowledge of the type of the noise) we would like to be able to decompose the time series into the deterministic trajectory and the measurement noise. Unfortunately this cannot be obtained directly from the embedded data, even when the dynamics are known *a priori*. To separate out the measurement noise from a time series requires the

relationship between dynamic errors and measurement errors to be explored. This relationship is set out in the next chapter and we then go on to investigate some ways in which we can use this to construct algorithms to determine measurement noise from dynamic noise.

3. Noise Reduction: Theory

In this chapter we investigate methods of reducing low amplitude noise from an underlying deterministic time series. At this stage we will assume that the data has already been embedded and that the dynamics are known *a priori*. Thus we have a 'noisy' trajectory that can be identified with some nonlinear dynamical system. To reduce the dynamic noise associated with the trajectory we aim to adjust the trajectory such that it is less noisy. This makes sense since the original trajectory can be interpreted as the new trajectory with additive measurement noise. Hence noise reduction, in this context, is a method of transforming dynamic noise into measurement error.

The idea for noise reduction of this type using state space methods was proposed by, among others, Kostelich and Yorke [1990]. Their method involved breaking the time series up into overlapping sections and adjusting these sections to make them more deterministic, while remaining 'close' to the original sections of the data. Unfortunately this does not provide a unique minimization process since it involves simultaneously minimizing the distance of the new trajectory from the old one and the distance of the data points from deterministic ones. Thus the cost function is a combination of these two aims:

$$S = \sum \left[w \| \hat{x} - x_i \|^2 + \| f(x_{i-1}) - x_i \| + \| f(x_i) - x_{i+1} \|^2 \right] \qquad (3.0.1)$$

where $\mathbf{f(x)}$ is the local mapping function, $\mathbf{x_k}$ is the original trajectory and $\hat{\mathbf{x}}_\mathbf{k}$ is the new 'cleaner' trajectory. The weighting $\mathbf{w}$ between the two parts is arbitrary. This arbitrariness is intrinsic in the problem of any noise reduction method and much of the work in this chapter looks into ways of solving this and other indeterminacies.

However in this chapter instead of solving the noise reduction problem proposed by Kostelich and Yorke we will concentrate on the problem of finding a completely deterministic orbit that is close to the noisy orbit, although later in section 3.4 we will relax this constraint slightly. The original work on finding 'close' deterministic orbits was done by Hammel [1990] whose method is based on the 'shadowing problem' of Bowen [1970,1978]. It is therefore useful to review Bowen's shadowing lemma before we consider Hammel's noise reduction algorithm.

In section 3.2 we return to the task of noise reduction and present the problem of finding a deterministic orbit close to the original data as a badly conditioned root finding problem. Hammel's algorithm, which was later called manifold decomposition, is then described in terms of a modified Newton-Raphson method. Then it is necessary to investigate the potential problems of applying this algorithm in practice. The major obstacle here is the presence of homoclinic tangencies. These manifest themselves by making the linearised problem singular, which in turn suggests that one solution may be to apply SVD. This is discussed in section 3.2.4 as well as some of its practical difficulties.

In section 3.4 the problem is reformulated as a minimisation problem. Although in many cases the minimum is equivalent to the zero in the root-finding problem and the problem is still badly posed it does allow more stable algorithms to be applied to reduce the noise. This approach is also compared to other methods that have recently been proposed in the literature. Although these new ideas have approached the problem from a different direction there is a great deal of similarity with the minimisation solution.

Finally to demonstrate the potential effectiveness of these methods a simple example of noise reduction is given applied to a time series from the Henon map. The mapping function is assumed to be known (the case where the function is approximated is dealt with in the next chapter) and the levels of noise reduction are shown to be impressive away from the regions near tangencies.

3.1 Pseudo-Orbits and Shadowing

The discussion at the end of the last chapter about the nature of the error term indicated that the relationship between dynamic error and measurement error needs to be explored in order to construct an algorithm that will separate a time series into true dynamics and measurement error. In fact much of the work required has already been done by Bowen [1970,1978], who was working on the problem of constructing Markov Partitions for *Axiom A* systems.

But before discussing this it is necessary to define certain terminology that is used in Bowen's work on the shadowing problem and will also help to put his work in the context of noise reduction algorithms.

Definition: **α-pseudo-orbit**: A sequence of points $\{x_i\}_{i=a}^{b}$ $x_i \in M$ for some mapping $f:M \rightarrow M$, will be described as an α-pseudo-orbit of f if it satisfies:

$$d(f(x_i),x_{i+1}) \leq \alpha \quad \forall \quad i \in [a,b-1] \tag{3.1.1}$$

where d is some measure of distance defined on M.

Definition: **β-shadow**: A point $y \in \Lambda$ is said to β-shadow a sequence $\{x_i\}_{i=a}^{b}$ if:

$$d(f^i(y),x_i) < \beta \quad \forall \quad i \in [a,b] \tag{3.1.2}$$

From these definitions a pseudo-orbit can be viewed as an approximation for an orbit with mapping errors at each point in the trajectory and an orbit that β-shadows it is a true orbit that is a close approximation to the pseudo-orbit. Thus the difference between the shadowing orbit and the pseudo-orbit can be interpreted as measurement noise. Therefore identifying a shadowing orbit for a given pseudo-orbit can be viewed as a noise reduction algorithm. Under suitable constraints this is what Bowen's proof of the Shadowing Lemma provides. The statement of the relevant theorem, which is slightly different to Bowen's is as follows:

Shadowing Lemma (Alekseev and Yakobson [1981]): *Let Λ be a hyperbolic set of $f:M \rightarrow M$. Then there is a neighbourhood $U \supset \Lambda$ such that for every $\beta > 0$, there is an $\alpha > 0$ such that every α-pseudo-orbit $\{x_i\}_{i=a}^{b} \subset U$ is β-shadowed by a point $y \in \Lambda$.*

The proof of this shadowing lemma is constructed by identifying the method by which a shadowing orbit can be found for a given pseudo-orbit. Since the explanation of the proof provides the basis for a noise reduction algorithm a sketch of it is presented below:

Sketch of the Proof

By the definition of a hyperbolic invariant set there exist c and $0<\lambda<1$ such that:

$$\begin{aligned} \|Tf^n|_{E^s}\| &< c\lambda^n \ , \ n\geq 0 \\ \|Tf^{-n}|_{E^u}\| &< c\lambda^n \ , \ n\geq 0 \end{aligned} \tag{3.1.3}$$

where E^s and E^u are defined as in chapter 1. This is a statement of the exponential expansion and contraction for the stable and unstable subspaces of the tangent space respectively. Similarly, as a direct consequence of the stable manifold theorem, this property can be extended from the infinitesimal to a local ϵ neighbourhood in the following way. There exists an ϵ such that:

$$\begin{aligned} y \in W^s_\epsilon(x),\ n \geq 0 &\ \Rightarrow\ d(f^n x, f^n y) \leq c\lambda^n d(x,y) \\ y \in W^u_\epsilon(x),\ n \geq 0 &\ \Rightarrow\ d(f^{-n} x, f^{-n} y) \leq c\lambda^n d(x,y) \end{aligned} \tag{3.1.4}$$

where $W^s_\epsilon(x)$ is the ϵ-neighbourhood of the local stable manifold of x and $W^u_\epsilon(y)$ is the ϵ-neighbourhood of the local unstable manifold of x, as defined in chapter 1. We also know from the stable manifold theorem that there exists a δ such that given x and y with $d(x,y) \leq \delta$ there is a unique point p by:

$$W^s_\epsilon(x) \cap W^u_\epsilon(y) = p \in \Lambda \tag{3.1.5}$$

Both these properties can be seen from the fact that these manifolds are locally a graph of the stable and unstable subspaces in the tangent space. Equation 3.1.5 is illustrated in Figure 3.1.1.

The basic idea behind the proof is to construct a sequence of orbits such that each new orbit will shadow more and more of the α-pseudo-orbit. This is achieved by choosing points that lie in intersections of stable and unstable manifolds as described above. Given ϵ and δ, as above, we can find a constant k such that:

$$\epsilon c\lambda^k < \delta/2 \tag{3.1.6}$$

This is required to guarantee that the kth iterate of the map provides a sufficient contraction forwards along the stable manifold and backwards along the unstable

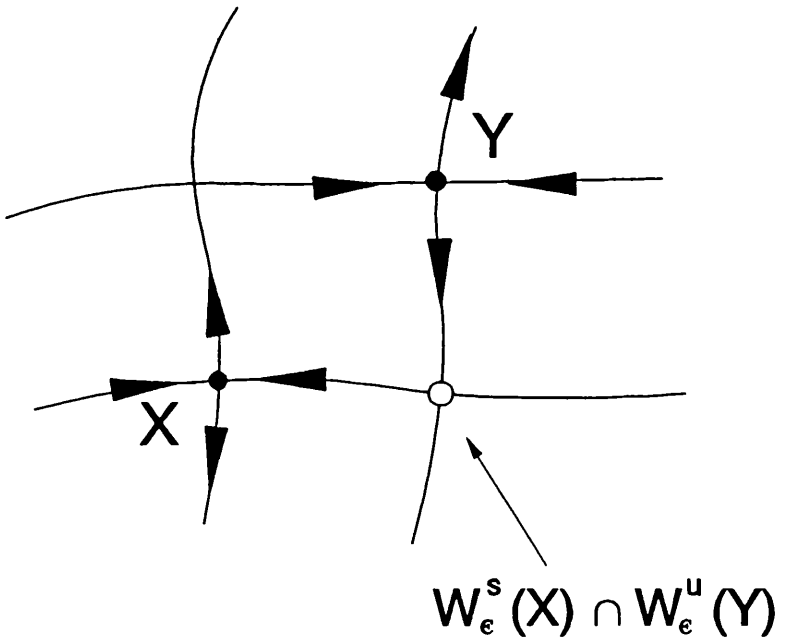


Figure 3.1.1: A graphical representation of the intersection of the local stable manifold of x and the local unstable manifold of y. This intersection can be shown to be unique in the ϵ-neighbourhood when x and y are less than δ apart.

manifold. Then there exists an α to satisfy:

$$d(f^j x_i, x_{i+j}) \leq \frac{\delta}{2} \quad , \quad 0 \leq j \leq k \tag{3.1.7}$$

Now we consider a finite $kn + 1$ length α-pseudo-orbit $\{x_i\}_{i=0}^{kn}$ and define a sequence of points $\{y_j\}_{j=0}^{n}$ setting $y_0 = x_0$ and then recursively setting:

$$y_i = W_\epsilon^s(x_{ki}) \cap W_\epsilon^u(f^k y_{i-1}) \tag{3.1.8}$$

This can only be defined if the distance between $f^k y_{i-1}$ and x_{ki} can be guaranteed to be within δ (equation 3.1.6). We prove this recursively by repeated application of the triangle inequality:

$$d(f^k y_{i-1}, x_{ki}) \leq d(f^k x_{k(i-1)}, x_{ki}) + d(f^k y_{i-1}, f^k x_{k(i-1)}) \tag{3.1.9}$$

So that if $d(y_i, x_{ki}) \leq \epsilon$ then $d(f^k y_{i-1}, f^k x_{k(i-1)}) \leq \epsilon c\lambda^k \leq \delta/2$ (from equation 3.1.4 and the definition of k) and $d(f^k x_{k(i-1)}, x_{ki}) \leq \delta/2$ (from our choice of α: equation 3.1.7). Therefore $d(f^k y_{i-1}, x_{ki}) \leq \delta$ and $d(y_{i+1}, x_{k(i+1)}) \leq \epsilon$ (from equation 3.1.5 and Figure 3.1.1). So, since $d(y_0, x_0) = 0$, we can extend this upper bound from for the distance between y_i and x_{ki} to all i.

$$d(y_i, x_{ki}) \leq \epsilon \tag{3.1.10}$$

We now claim that $f^{-kn} y_n$ will β-shadow the pseudo-orbit for some β. This is because y_n lies in the unstable manifolds of all y_i, $0 \leq i \leq n$ and therefore we expect the distance from x_{ki} and $f^{k(n-i)} y_n$ to be bounded. This can be shown as follows:

$$\begin{aligned} d(x_{ki}, f^{k(n-i)} y_n) &\leq d(y_i, x_{ki}) + \sum_{j=0}^{n-i-1} d(f^{-jk} y_{i+j}, f^{-(j+1)k} y_{i+j+1}) \\ &\leq \epsilon + \epsilon c\lambda^k + \epsilon c\lambda^{2k} + \epsilon c\lambda^{3k} \ldots \end{aligned} \tag{3.1.11}$$

which is a geometric sum and since $\|\lambda\| \leq 1$ this sum converges:

$$d(x_{ki}, f^{-k(n-i)}) \leq \epsilon + \frac{\epsilon c\lambda^k}{1 - \lambda^k} \tag{3.1.12}$$

Finally to show shadowing for the complete trajectory we must include the points x_i where i is not divisible by k but we have already bounded this distance in equation 3.1.7. Thus to include these points merely requires an additional $\delta/2$ and we can bound β in

terms of δ and ϵ from the stable manifold theorem as follows:

$$\beta \leq \delta/2 + \epsilon + \frac{\epsilon c \lambda^k}{1 - \lambda^k} \qquad (3.1.13)$$

This proves shadowing for any finite length pseudo-orbit. To extend this argument to a bi-infinite pseudo-orbit it is sufficient to consider a sequence of points that shadow finite segments of it. Since the hyperbolic set is compact there exists a limit of some sequence of these points that shadows the whole pseudo-orbit.

A further corollary to this theorem is that the shadowing orbit for a bi-infinite pseudo-orbit is unique. This is obviously necessary when trying to find a true orbit associated with noisy data. Although, in reality, this uniqueness does not hold for finite length orbits a sensible implementation of the process will circumvent this problem. Further discussion of this is postponed until sections 3.2 and 3.3.

Before detailing the implementation of practical algorithms it is important to consider some of the limitations of adopting the shadowing approach to noise reduction. The major problem of the Shadowing Lemma is that it can only be applied to hyperbolic systems, although it can be extended to cover any hyperbolic trajectory. Indeed, in general, pseudo-orbits from non-hyperbolic systems are unshadowable. This is because whenever a ϵ-pseudo-orbit approaches within ϵ of a tangency (or even a near-tangency: see Figure 3.1.2) there is a chance that it will cross the separatrix thus making it impossible to shadow. Whereas simple examples of non-hyperbolicity are non-generic it is believed that many chaotic attractors contain homoclinic tangencies and hence are non-hyperbolic. Furthermore the presence of homoclinic tangencies in an invariant set can be shown to persist under perturbation (see Guckenheimer and Holmes [1983] and the references therein).

Although this puts a severe limitation on finding shadowing trajectories for pseudo-orbits in general chaotic attractors it was pointed out by Hammel et al [1988] that the ideas in the shadowing lemma can still be applied to finite lengths of trajectory as long as they are sufficiently hyperbolic. Whereas the method in the proof of the Shadowing Lemma, given above, is not directly applicable as a stable algorithm, the basic ideas of splitting

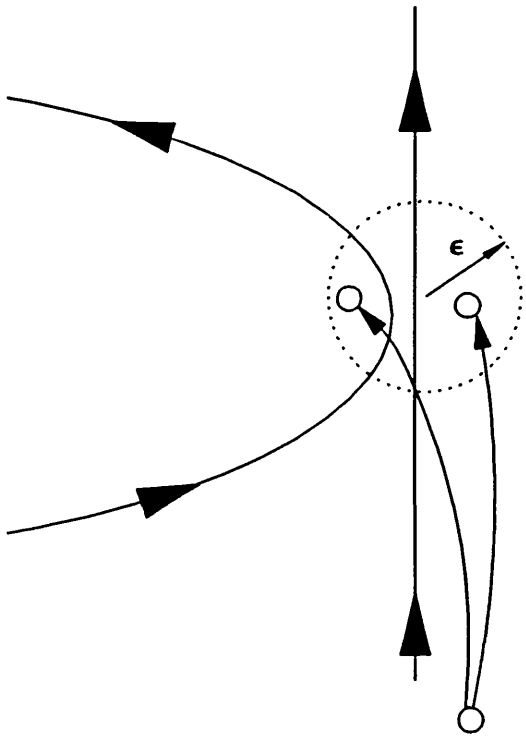


Figure 3.1.2: An ϵ-pseudo orbit is unshadowable if the error is large enough to cross over a stable manifold. This is likely to happen for any ϵ in a chaotic attractor that is not strictly hyperbolic if the trajectory is long enough.

the space into stable and unstable subspaces was used by Hammel et al to practically implement a shadowing algorithm to numerically prove the existence of nearby true orbits to numerically calculated ones.

Unfortunately Hammel's algorithm requires very low levels of noise whereas in practical noise reduction an algorithm should be robust to realistic noise levels. However the mere application of a noise reduction algorithm, in general, implies the belief that the noise is measurement noise and not dynamic error and this in turn implies that the resulting pseudo-orbit automatically has a shadowing orbit. In such a situation implementation of Hammel's algorithm can still lead to problems since if the pseudo-orbit is not hyperbolic then Hammel's algorithm becomes unstable even though a true shadowing orbit exists. The remaining sections of this chapter investigate practical solutions to these problems in order to produce a noise reduction algorithm that is robust to large noise levels and the ill-conditioning introduced by non-hyperbolicity.

3.2 Zero-Finding and the Shadowing Problem

Before any noise reduction can be considered it is necessary to make several definitions. First it is assumed that the time series originates from a low dimensional deterministic dynamical system and that the time series has been reconstructed in some suitable state space to produce a vector series of n points in state space where x_i is defined as the ith point in the series. The deterministic mapping function is then approximated in some manner, if it is not known *a priori*, and the approximation errors are defined as

$$\epsilon_i = x_i - f(x_{i-1}) \qquad \textbf{(3.2.1)}$$

where ϵ_i is the approximation error associated with the ith point in the series and $f()$ is the mapping function.

The aim then is to remove the approximation errors from the orbit. Unfortunately in general the measurement errors cannot be directly decomposed from the observed data. However if we consider the error function as a function of the trajectory then the shadowing problem becomes one of finding the adjusted trajectory for which the error

function is zero. This approach to noise reduction is equivalent to solving an n-dimensional root finding problem and the obvious choice of algorithm by which to do this is to use a Newton-Raphson method. This is well known and commonly found in numerical procedures. It is based on linearizing the function as an estimate for the position of the root. If the root is approximated by its Taylor expansion it can be assumed to be a linear function of the initial guess the position of the root can be determined directly:

$$x_{new} = x_{cur} - \left(Dg(x_{cur})\right)^{-1} g(x_{cur}) \tag{3.2.2}$$

where $g(x)$ is the error function in this case and $Dg(x)$ is the derivative with respect to x. As long as the initial guess is close enough to the zero the algorithm will converge onto it at a quadratic rate, see for example Press et al [1992].

To apply it to the noise reduction problem the error function, given in equation 3.2.1, must first be linearised around the solution and set to zero:

$$x_i + \Delta x_i - f(x_{i-1}) - J_{i-1}\Delta x_{i-1} = 0 \tag{3.2.3}$$

where J_i is the Jacobian of the mapping function $f(x_i)$ at the ith data point. Then substituting equation 3.2.1 back in we have:

$$\Delta x_i - J_{i-1}\Delta x_{i-1} = -\epsilon_i \tag{3.2.4}$$

This can be written in matrix form in the following way:

$$D\,(\underline{x}_{new} - \underline{x}_{cur}) = -\underline{\epsilon}$$

$$D = \begin{bmatrix} 1 & -J_1 & & & & \\ & 1 & -J_2 & & & \\ & & 1 & -J_3 & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & 1 & -J_{n-1} \end{bmatrix} \tag{3.2.5}$$

It is immediately apparent from equation 3.2.5 that to solve this for x_{new} requires inverting the matrix D which is not possible since D is not a square matrix but instead is an $n\text{-}1 \times n$

matrix with d-dimensional elements, where d is the dimension of the reconstructed state space. This is because the number of data points is greater than the number of error terms (this comes back to the non-uniqueness of solution mentioned at the end of section 3.1). It can only be solved by adding extra constraints. One method of applying a reasonable set of additional constraints is to constrain the ends in the stable and unstable subspaces. This was proposed by Hammel [1990] and is briefly reviewed below.

3.2.1 Manifold Decomposition

Hammel [1990] devised a method for noise reduction by finding a deterministic trajectory that shadows the whole data set. The approach used was based on earlier work on the shadowing problem, Hammel *et al* [1988], in which they proved that finite lengths of numerical approximations to the Henon map actually shadowed real trajectories to within a small distance.

The algorithm to find the shadowing orbit assumes that the dynamics can be linearized around the time series embedded so that a Newton-Raphson type approach can be adopted. If we start from equation 3.2.4 it is clear that one of the zero solutions for this equation can then be obtained by initially choosing an arbitrary value for Δx_1 (e.g. zero) and then solving for the remaining Δx_i iteratively. Of course this solution would only be a sensible one as long as the errors in the initial choice of Δx_1 are not amplified under iteration: this would invalidate the linearisation assumption used in Newton-Raphson.

Unfortunately if the trajectory is chaotic then along the unstable directions this solution will be unstable. However this can be solved by decomposing the problem into stable and unstable subspaces and setting Δx_1^s and Δx_n^u equal to zero. Then as long as the subspaces span the full linear space the iterative procedure can be applied forwards along the stable direction and backwards along the unstable direction, resulting in the following recursive solutions:

$$\Delta x_{i+1}^s = J_i^s \Delta x_i^s - \epsilon_{i+1}^s \qquad \textbf{(3.2.6)}$$

and

$$\Delta x_i^u = \left(J_i^{-1}\right)^u (\Delta x_{i+1}^u + \epsilon_{i+1}^u) \qquad (3.2.7)$$

where the superscripts s and u denote the stable and unstable subspaces respectively. The resulting adjusted orbit will then be deterministic to within the limits of the local linear assumption. The process can then be repeated several times in the usual way to improve the accuracy of the estimate of the shadowing orbit.

3.2.2 Approximation of the Stable and Unstable Manifolds

The method of manifold decomposition cannot be implemented without first dividing the local neighbourhoods in state space into stable and unstable subspaces. Unfortunately however it is not practical to try to calculate the actual manifolds for each neighbourhood, since this requires the knowledge of the forward trajectory $t \to \infty$ and the backward iterated trajectory, $t \to -\infty$. In fact the exact decomposition is not necessary and it is possible to approximate this decomposition in a way that is sufficient for the above algorithm. Manifold decomposition requires the local tangent spaces of the data to be decomposed into two invariant subspaces such that adjustments restricted to the stable subspace contract on average when iterated forwards along the time series. Similarly adjustments restricted to the unstable subspace contract on average when iterated backwards.

If the dynamical system is only two-dimensional then we can use the fact that generically the angle between any arbitrary vector in the local tangent space iterated forwards along the time series and the direction of the maximal expansion will reduce and tend to zero in the limit $t \to \infty$. Thus, this method provides a good approximation of the unstable directions along the time series. Similarly an arbitrary choice of stable direction can be made for the last point in the time series and iterated backwards to approximate the local stable manifold.

For higher dimensional data sets it is necessary to calculate the directions associated with the Lyapunov exponents since the directions associated with the positive Lyapunov

exponents span the unstable manifold and similarly the negative Lyapunov directions span the stable manifold. Therefore to implement manifold decomposition in higher dimensional systems we can turn to the methods for calculating Lyapunov exponents, as described in chapter 1.

Consider the factorisation of a sequence of matrices, $J_i \in \mathbf{R}^d$, that was described in section 1.4 into orthogonal matrices Q_i and upper triangular matrices with positive diagonals, R_i. Setting Q_0 to the identity, we have:

$$\begin{aligned} J_1 Q_0 &= Q_1 R_1 \\ J_2 Q_1 &= Q_2 R_2 \\ &\cdots \\ J_i Q_{i-1} &= Q_i R_i \end{aligned} \qquad \textbf{(3.2.8)}$$

such that the product of these matrices can be written as:

$$\prod_{i=1}^{n} J_i = Q_n R_n R_{n-1} \cdots R_1 \qquad \textbf{(3.2.9)}$$

Since the all R_i are upper triangular the first k column vectors of Q_i provide a k-dimensional subspace that is invariant under the mapping J_{i+1}:

$$J_{i+1} Q_i(1 \to d, 1 \to k) = Q_{i+1}(1 \to d, 1 \to k) R_{i+1}(1 \to k, 1 \to k) \qquad \textbf{(3.2.10)}$$

where the notation $Q(1 \to d, 1 \to k)$ means the submatrix with the elements q_{ij} for $1 \le i \le d$ and $1 \le j \le k$. Since this subspace will span a k-dimensional subspace tending onto the k most expanding directions we can identify the dimension of the local unstable manifold, d_u, (i.e. the number of Lyapunov exponents greater than zero) and then the invariant subspace spanned by the first d_u column vectors in each Q_i provides an approximation for the unstable manifold at x_i.

Unfortunately the remaining column vectors in Q_i do not provide an invariant basis for the local stable manifold at x_i since they are constrained to be orthogonal to the unstable subspace and are therefore not invariant. This is not really surprising since the local stable manifold is only defined in terms of the limit of backwards iterates of J_i^{-1} and we can consider a backwards factorisation of J_i with the same form as equation 3.2.8:

$$
\begin{aligned}
&Q_n = I \\
&J_{n-1}^{-1} Q_n = Q_{n-1} R_{n-1} \\
&J_{n-2}^{-1} Q_{n-1} = Q_{n-2} R_{n-2} \\
&\cdots
\end{aligned}
\qquad (3.2.11)
$$

Thus we can calculate a basis for an invariant d_s-dimensional subspace ($d_s = d - d_u$) and in a similar way to the construction of the unstable subspace this basis will span the d_s most expanding directions with respect to the mappings J_i^{-1}. Therefore this approximates the stable manifold as $i \to 1$.

Although this method will not accurately approximate the stable subspace at the end of the time series and the unstable subspace at the beginning of the time series this is not important. In manifold decomposition the requirements of the stable and unstable subspaces is that the mapping restricted to the stable subspace typically contracts and the mapping restricted to the unstable subspace typically expands. The decomposition method described here will in general satisfy these conditions.

3.2.3 Tangencies Imply Ill-conditioning of D

It has already been mentioned that many chaotic attractors are likely to be non-hyperbolic and in a non-hyperbolic attractor almost all bi-infinite orbits will pass arbitrarily close to some given tangency. Since in general a pseudo-orbit from a non-hyperbolic system cannot be shadowed by a true orbit we can expect the presence of tangencies or near-tangencies to make the zero shadowing problem ill-conditioned (as well as being rank deficient) and solutions using the method of manifold decomposition are liable to fail. This manifests itself is the form of the matrix D in equation 3.2.5 becoming singular, as we will show below.

In this context our definition of a near-tangency in the linearised problem is as follows:

Definition **ϵ-near-tangency**. There exists an ϵ-near-tangency at the mth point in a pseudo-orbit if we can find unit vectors u_m, in the

unstable subspace, and s_m, in the stable subspace such that
$s_m.u_m = cos\theta$ and $\theta < \epsilon$.

where the unstable and stable subspaces here are defined by the directions associated with the singular values of J^n that are greater than or less than one respectively. Initially we will find a lower bound for the condition number of D, with the added constraints, $\Delta x_1^s = \Delta x_n^u = 0$, as a function of the angle between the directions s_m and u_m at some position m in the trajectory.

First we can show that D has at least one singular value greater than or equal to one: i.e. there exists an Δx such that $\|\Delta x\| \leq \|e\|$, where $e = D\Delta x$. Choose Δx such that $\Delta x_i = 0$, if $i \neq m$ and $\Delta x_m \neq 0$. Then we can calculate e from equations 3.2.6 and 3.2.7: $e_i = 0$, if $i \neq \{m, m+1\}$, $e_m = -\Delta x_m$ and $e_{m+1} = J_m e_m$. Since $\|e_m\| = \|\Delta x_m\|$ we know that $\|\Delta x\| \leq \|e\|$ and thus D has a singular value greater than or equal to one.

Now given an ϵ-near-tangency at the mth point we can construct an upper bound for the smallest singular value. To do this it is necessary to find an e such that there is a constant $K \geq \|\Delta x\| / \|e\|$ for all Δx that satisfy $D\Delta x = e$. In terms of the solution by manifold decomposition this means: for all choices of Δx^s_1 and Δx^u_n.

Let us choose e such that $e_i = 0$, if $i \neq m$ and $e_m = \alpha u_m + \beta s_m$. Initially let $\Delta x^s_1 = \Delta x^u_n = 0$. Then we can solve for Δx using the method outlined in section 3.2.1. Thus we know that:

$$\begin{aligned} \Delta x_i^s &= 0 \quad , \quad i < m \\ \Delta x_i^u &= 0 \quad , \quad i \geq m \\ \Delta x_m^s &= -e_m^s \end{aligned} \tag{3.2.12}$$

and $\|\Delta x\| \geq \|e_m^s\| = \|\beta\|$. If we now choose $e_m.s = 0$ we can write:

$$\begin{aligned} \alpha &= e_m.\underline{u} - \beta\cos\theta \\ \beta &= \alpha\cos\theta \end{aligned} \tag{3.2.13}$$

and since $e_m.s = 0$, and $u.s = cos\theta$ then $e_m.u = \|e_m\| sin\theta$ and we can write β in terms

of $\|e_m\|$:

$$\beta = \left\lceil \frac{\|e_m\| \sin\theta}{(1 + \cos^2\theta)} \right\rceil \tag{3.2.14}$$

and by construction $\|e\| = \|e_m\|$. Therefore we have:

$$\frac{\|\Delta x\|}{\|e\|} \leq \left| \frac{\sin\theta}{(1 + \cos^2\theta)} \right| \leq \epsilon \tag{3.2.15}$$

Thus, under the assumptions set out above, we have shown that the condition number of D with the additional constraints is greater than $1/\epsilon$.

To consider the condition number of D without constraints would require us to demonstrate that Hammel's method is ill-conditioned for any constraints on Δx_1^s and Δx_n^u chosen. However, to extend this argument to any constraints applied to D is more difficult. Here we merely provide a heuristic explanation of why D is ill-conditioned. If we attempted to select a value for Δx_1^s that kept $\|\Delta x_m\| / \|e\|$ small this would require Δx_1^s to be of the order of $1/\epsilon$ since:

$$\Delta x_m^s = \prod_{i=1}^{m} J_i \Delta x_1^s - e_m^s \tag{3.2.16}$$

and we expect vectors in the stable subspace to contract when mapped forwards. Thus the condition number would still be large since $\|\Delta x_1\| / \|e\|$ would be large. A similar argument can be constructed for the selection of Δx_n^u and indeed for any combination of the two (this can be done since we chose the stable and unstable subspaces such that they were mutually orthogonal at the ends of the trajectory).

3.2.4 Solution by SVD

Since tangencies appear to be common in chaotic systems it is worth considering now best to deal with the singularities they induce in the matrix **D**. The most robust method for solving badly conditioned problems is to use singular value decomposition and the Moore-Penrose pseudo-inverse. If we treat the small singular values as exactly zero this

will result in a k-dimensional family of solutions for the problem $Dx = e$, where k is the number of small singular values. Then the problem becomes one of choosing the *best* solution for our particular needs. This can be achieved by applying the Moore Penrose pseudo-inverse. Let $D = USV^T$ be the singular value decomposition where U and V are orthonormal matrices and S is a diagonal matrix whose elements are the singular values of D. Then it is simple to write the solution to $Dx = e$ in terms of U, S, and V:

$$x = eVS^{-1}U^T \tag{3.2.17}$$

where S^{-1} is just the diagonal matrix whose elements are $1/s_{ii}$. This immediately identifies the problem of a badly conditioned matrix since if $s_{ii} \to 0$ then $1/s_{ii} \to \infty$. The Moore-Penrose pseudo-inverse replaces $1/s_{ii}$ by 0 whenever $s_{ii} = 0$ (or in our case whenever s_{ii} is numerically small). This method makes sense since it chooses the point from the family of solutions that minimises the L_2 norm of $\|x\|/\|e\|$ and thus chooses the smallest solution that satisfies the problem. In fact, if this method for inverting D is used, the manifold decomposition is redundant since it was a mechanism for applying stable constraints to overcome the problem of D not being a square matrix and hence there being no unique solution for the problem. As we have seen above this problem is automatically solved by the application of the Moore-Penrose pseudo-inverse.

Unfortunately, although a solution by singular value decomposition is very attractive it is not always a practical option. One of the major benefits of manifold decomposition is its speed: it is an order N process. This is because it is able to exploit the banded structure of D. However singular value decomposition does not have a fast banded algorithm since U and V will generally have no zero elements. This means we cannot make the process faster than order N^3 which is the standard process time for singular value decomposition. Thus the application of this method to the problem will soon become impractical if the data length used becomes too large.

Obviously we can resort to compromise solutions using this method. For example, in the same way as Kostelich and Yorke [1990] broke up their data into small overlapping sets, we could apply singular value decomposition repeatedly to small lengths of data, however it then becomes a problem to choose how large these lengths of data should be, since the smaller the length of the data is the less effective the resulting noise reduction is. Another

compromise would be to apply manifold decomposition where the decomposition is stable and to apply singular value decomposition in the regions where tangencies are a problem. This method was proposed by Farmer and Sidorowich [1991] for the solution of a slightly different problem (see next section).

3.3 Implicit and Explicit Shadowing.

At the end of section 3.1 we mentioned that finite length pseudo-orbits do not possess unique shadowing orbits. Instead there is a d-dimensional family of shadowing orbits, where d is the dimension of the state space. In terms of Hammel's method this set can be realised by perturbing the end constraints in the stable and unstable directions. Therefore if we chose a non-zero end constraint we will obtain a different shadowing orbit.

This poses the question: *can we do any better?* The answer is, theoretically, yes we can. To start with the solution given in the last section finds the closest (in the L_2 norm) estimate for the shadowing orbit at each iterate of the linearised equations but even here the is an implicit assumption that optimising each linear step provides a good solution to the global nonlinear problem. A better approach would be to explicitly search for the closest shadowing orbit to the original pseudo-orbit. This was proposed by Farmer and Sidorowich [1991] and was rightly called 'optimal shadowing' since in finding the deterministic orbit that is closest to the pseudo-orbit it can be viewed as a maximum likelihood estimate. They applied this by minimising the following cost function:

$$S = \sum_{k=1}^{N} \| y_k - x_k \| + 2 \sum_{k=1}^{N-1} [f(x_k) - x_{k+1}]^T \lambda_k \qquad \textbf{(3.3.1)}$$

where $\mathbf{f(x)}$ is the mapping function as before, $\mathbf{x_k}$ is the deterministic trajectory, $\mathbf{y_k}$ is the noisy trajectory and $\boldsymbol{\lambda_k}$ are the Lagrange multipliers. Farmer and Sidorowich linearised this function and solved the approximation iteratively using the Newton-Raphson method. The resulting linearised equations are:

$$\begin{aligned} \gamma_n &= J_n \lambda_n - \lambda_{n-1} - \delta x_n \\ \delta x_{n+1} &= J_n \delta x_n + \epsilon_n \end{aligned} \qquad \textbf{(3.3.2)}$$

where the new variables are defined as:

$\hat{x}_n$	=	present estimate for x_n
$\epsilon_n = \hat{x}_{n+1} - f(\hat{x}_n)$	=	approximation error
$\gamma_n = y_n - \hat{x}_n$	=	deviation from the noisy orbit
$\delta x_n = \hat{x}_n - x_n$	=	deviation from new estimate for x_n

This can be written in matrix form as:

$$Mv = w$$

$$\begin{bmatrix} -1 & J_1 & & & & & & \\ J_1 & 0 & -1 & & & & & \\ & -1 & -1 & J_2 & & & & \\ & & & \cdot & & & & \\ & & & & \cdot & & & \\ & & & & & \cdot & & \\ & & & & & J_{n-1} & 0 & -1 \\ & & & & & & -1 & -1 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \lambda_1 \\ \delta x_2 \\ \lambda_2 \\ \\ \\ \lambda_{n-1} \\ \delta x_n \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \epsilon_1 \\ \gamma_2 \\ \epsilon_2 \\ \\ \\ \epsilon_{n-1} \\ \gamma_n \end{bmatrix} \qquad \textbf{(3.3.3)}$$

Since optimal shadowing provides a unique solution even for finite length data sets the rank deficiency of D is removed by solving the problem with M which is square. However in Farmer and Sidorowich, it was conjectured that the matrix M would be badly conditioned if the data came from a chaotic system. This can be shown to be incorrect by considering the following simple counter example. If the data came from the *2x mod 1* map then the Jacobian of the system would be constant and the matrix M would be:

$$M = \begin{bmatrix} -1 & 2 & & & & & & \\ 2 & 0 & -1 & & & & & \\ & -1 & -1 & 2 & & & & \\ & & & \cdot & & & & \\ & & & & \cdot & & & \\ & & & & & \cdot & & \\ & & & & & 2 & 0 & -1 \\ & & & & & & -1 & -1 \end{bmatrix} \qquad \textbf{(3.3.4)}$$

The condition number for this matrix remains small for even large values of n and so the matrix is not ill-conditioned. Figure 3.3.1 shows a plot of the numerical calculated condition number of this matrix for $n < 50$ and it appears that the condition number is

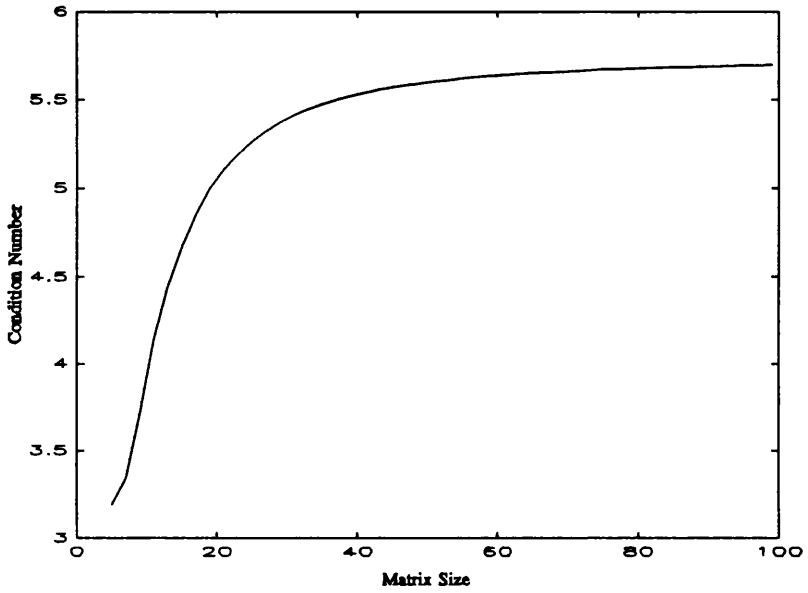


Figure 3.3.1 : A plot of the condition number of matrix M for the $2x \ mod \ 1$ map against size of the matrix. It is clear that the condition number does not grow exponentially with the matrix size.

not likely to increase much beyond 6 for large n. Certainly this is not the exponential growth of the condition number predicted. Although the presence of chaos does not necessarily introduce ill-conditioning into M this does not mean that manifold decomposition would not be necessary if the data came from a hyperbolic chaotic attractor. It is clear from equation 3.3.3 that M is not diagonally dominant since it has zeros along the diagonal. Therefore if standard methods of matrix inversion are used to solve this equation, such as, Gaussian elimination, pivoting will be required. This means that it would not be possible to exploit the banded structure of the matrix and a straight forward approach to solving $Mv = w$ would not result in an order N process. Here manifold decomposition can be used, not to stabilise an ill-conditioned matrix, but to provide a faster solution.

The ill-conditioning that Farmer and Sidorowich observed was, in fact, due to the presence of homoclinic tangencies. At first it might appear that the minimisation process could remove these singularities in the same way that it does for the rank deficiency described above but there is a fundamental difference. The singularities in the matrix D were due to the fact that it was a non-square matrix and therefore had singular values that were zero, whereas generically a finite length orbit from a dynamical system will not have any exact tangencies. Instead near-tangencies will result in some singular values becoming very small but while the singular values remain strictly positive the linearised equations still have an exact solution (however absurd this is). This is a common predicament in least squared estimation since a problem can be ill-conditioned even when a set of equations may be over-determined.

To stabilise their scheme Farmer and Sidorowich proposed a joint application of manifold decomposition and SVD. They proposed identifying approximations to the stable and unstable subspaces for the trajectory and then finding any near-tangencies present. Then away from near-tangencies manifold decomposition can safely be applied whereas in the regions of the tangencies SVD can be applied.

It is clear that the formulating the problem in terms of explicit shadowing has not made the algorithm any more stable in the presence of tangencies therefore it is necessary to consider what benefits will be gained by this additional work. Obviously it is likely to

find an orbit that is closer to the pseudo-orbit than the methods of manifold decomposition, but how much closer is "closer"? This can be determined by decomposing the problem into the method of Hammel and the optimisation of the initial constraints of δx_1^s and δx_n^u (i.e. a general elimination method: see section 4.3 for more details). If the system has well defined Lyapunov exponents then perturbations applied to the initial conditions will introduce changes predominantly at ends of the shadowing orbit and these changes will tend to decrease exponentially. The only other points that are badly defined are the tangencies due to the problem being badly conditioned. Also since most of the orbit is "nearly" unique except at the ends and the homoclinic tangencies the maximum likelihood estimate is only using information local to these bad points and is therefore not going to produce a statistically significant improvement. This emphasises the importance of the uniqueness of the shadowing orbit (and the convergence of finite length orbits onto it as the length tends to infinity). It is this fact and not a statistical argument that makes chaotic noise reduction so powerful.

3.4 Minimising Noise and Weak Shadowing

In this section the problem is reformulated as a minimisation problem. This is a less ambitious aim and is not confined to the assumption that a deterministic orbit and the original noisy orbit are approximately within the same linear neighbourhood because the minimisation problem merely searches for a *less* noisy orbit close to the original noisy data. One consequence of this is that a minimisation algorithm will not necessarily be forced to be unstable in situations where there are no close true orbits as is the case when homoclinic tangencies occur in the dynamics.

In the minimisation problem the aim becomes one of minimising the following cost function.

$$H = \sum_{i=2}^{n} \epsilon_i^2 \qquad \textbf{(3.4.1)}$$

where the error term is the same as that defined earlier in equation 3.2.1. Two methods are discussed below to solve the n-dimensional minimisation problem. The first method

is a solution by gradient descent and the second approach is a compromise between the gradient descent method and the minimisation analogue of the Newton-Raphson algorithm.

3.4.1 Solution by Gradient Descent

The simplest and most robust method for minimising a function is to take small steps 'down hill'. This is best achieved using a steepest descent method. One major advantage of this over the Newton method is that it makes no assumption of the form of the cost function and merely requires a smoothness criterion. The basic algorithm is:

$$x_{new} = x_{cur} - constant \times \nabla H(x) \tag{3.4.2}$$

where the constant defines the size of the step. ∇H can be determined by differentiating equation 3.4.1.

$$\frac{\partial H}{\partial x_k} = 2 \sum_{i=1}^{n} \frac{\partial \epsilon_i}{\partial x_k} \tag{3.4.3}$$

This evaluates the partial gradient for the kth point in the state space. Although equation 3.4.3 has to be evaluated for each point in the state space this is not difficult, since the definition of the error function in equation 3.2.1 means that virtually all the terms in the summation are zero:

$$\frac{\partial \epsilon_i}{\partial x_k} = 0 \ , \ i \neq k, \ k+1 \tag{3.4.4}$$

Therefore equation 3.4.3 can be rewritten as:

$$\frac{\partial H}{\partial x_k} = \epsilon_k - J_k \epsilon_{k+1} \tag{3.4.5}$$

where J_k is the Jacobian of the approximating function $f(x_k)$ at the kth point. ∇H can therefore be calculated in order n operations and the routine can be iterated until the level of noise is low enough.

It is interesting that the gradient descent approach does not suffer from the fact that the

problem is ill-posed or the ill-conditioning that will be introduced by homoclinic tangencies, thus no extra constraints need to be added. This is essentially because these singularities in D are equivalent to directions on the cost function that are flat, therefore the gradient descent method of equation 3.4.2 will always go perpendicular to the singular directions (it is no coincidence that $\nabla H = -D^T\epsilon$ and hence the gradient descent direction projects out the singular directions in D).

However homoclinic tangencies do still cause some problems when using a gradient descent algorithm since when the gradient of the cost function becomes almost flat repeated iterations of the algorithm cease to reduce the error at a linear rate. This was also observed numerically by Grassberger et al [1992]. However it is infinitely more preferable for a noise reduction algorithm to grind to a halt at tangencies than for it to become unstable.

It is worth briefly mentioning that there is nothing particularly magical about the least squared norm and any other norm would be equally valid for the error function given in equation 3.4.1. Obviously whatever function is chosen should not require excessive computation but within this restriction there are alternatives to the Euclidean norm that may have some advantages in terms of their convergence rates. A gradient descent method based on the least squared error will converge to a minimum at a rate approximately proportional to the size of the error. This convergence rate could easily be improved by simply using a norm with a faster convergence. For example Zak [1989] proposed minimising:

$$\sqrt[3]{\sum_{i=1}^{n} e_i^4} \qquad \textbf{(3.4.6)}$$

in a neural network problem. However changing the norm will change the size of the maximum step that can be taken without losing stability and this may well limit the extent to which any improvements in speed can be made.

3.4.2 A Comparison with Other Methods

It is convenient, at this point, to compare the ideas presented in the last section to some other schemes that have been proposed for noise reduction algorithms. The methods of Hammel [1990] and Farmer and Sidorowich [1991] have already been reviewed in some detail above, but recently some new ideas for stable noise reduction algorithms in the presence of large noise levels have been proposed by Sauer [1992], Cawley and Hsu [1992] and Schreiber and Grassberger [1992]. The aim of these new methods is to be able to filter data when the noise levels are too high to be able to successfully apply the methods of Hammel or Farmer and Sidorowich. We have already shown that in the context of the minimisation problem presented above stable noise reduction for high noise levels can be achieved by taking small enough steps down the cost function. In this section we will demonstrate that these new ideas can be considered in terms of minimisation and they can be classed as generalised gradient descent methods.

The algorithms presented by Cawley and Hsu and Sauer are very similar and we will concentrate on the method of Cawley and Hsu here. Although these algorithms incorporate advanced methods for embedding the data (global and local singular systems analysis) if we assume that the method of noise reduction can be evaluated independently from the function approximation and the embedding (this assumption is discussed in detail in chapter 4) then we can compare the algorithm of Cawley and Hsu directly to the method of gradient descent.

Cawley and Hsu apply local singular systems analysis to the data to produce local neighbourhoods that linearly map the state vector x_i to x_{i+1}. This is a local linear approximation for the mapping function. Then, having identified the function locally, the data can be projected down onto, or at least towards, the graph of the map from $\boldsymbol{R}^d \rightarrow \boldsymbol{R}^d$, such that the error is reduced for each pair of points in the trajectory. However, given a linear co-dimension 1 subspace in $\boldsymbol{R}^{d \times d}$, there are many directions in which the data can be projected onto the surface but the natural direction is orthogonal and this is the one that is used (this is illustrated in Figure 3.4.1). The result of this projection is two new values for each point, one for the when the point is in the domain and one for when it is in the range:

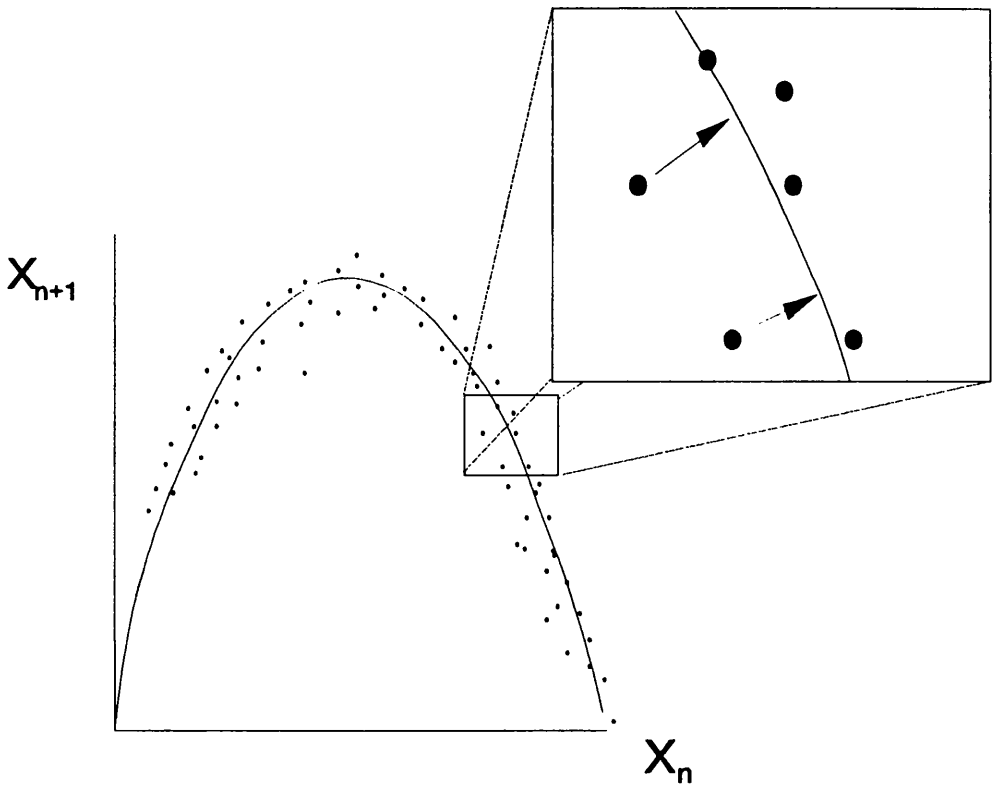


Figure 3.4.1 : A graphical representation of the projection noise reduction method of Cawley and Hsu [1992] and Sauer [1992]. This method can be shown to be equivalent to noise reduction by gradient descent.

$$x_{i,new} = x_{i,old} - \delta.\epsilon_{i-1} \tag{3.4.7}$$

and

$$x_{i,new} = x_{i,old} + \delta.J_i\epsilon_i \tag{3.4.8}$$

where J_i is the linear mapping function and δ is a small constant defining the extent to which the points are adjusted towards the corrected values. Therefore in this approach a compromise has to be sought to obtain a single consistent orbit. This is achieved by averaging the two corrections:

$$x_{i,new} = x_{i,old} - \delta.(\epsilon_{i-1} - J_i\epsilon_i) \tag{3.4.9}$$

A comparison of this equation to equation 3.4.2 makes it immediately obvious that the method of Cawley and Hsu (and similarly Sauer's method) is a reformulation of the steepest descent method for the cost function given in equation 3.4.1.

Finally the steepest descent approach can be compared to an algorithm proposed by Schreiber and Grassberger ([1991] and Grassberger et. al. [1992]). This method makes similar small adjustments to the trajectory to reduce the overall dynamic error. However the adjustment is just delta times the approximation error:

$$x_{i,new} = x_{i,old} + \delta\ \epsilon_i \tag{3.4.10}$$

If this error was the same as that defined in equation 3.2.1 then the resulting noise reduction would only take place in the stable direction, therefore they used a function approximation that contained information from both the past and the future, for example:

$$x_i = F(x_{i-d1},x_{i-d1+1}..,x_{i-1},x_{i+1},...x_{i+d2}) \tag{3.4.11}$$

where *d1* and *d2* are the dimensions of the past and future parts of the delay vector. In such a case the steepest descent adjustment for the error function would be:

$$\frac{\partial H}{\partial x_i} = -\frac{\partial F_{i-d1}}{\partial x_i}\epsilon_{i-d1} - \ldots \frac{\partial F_{i-1}}{\partial x_i}\epsilon_{i-1} + \epsilon_i - \ldots \frac{\partial F_{i+d2}}{\partial x_i}\epsilon_{i+d2} \tag{3.4.12}$$

where everything is defined as above. The argument for just using the error term is that since the function approximation is non-predictive the additional terms will on average

be small and can therefore be ignored. This was proved for a simple case using the Baker map, Schreiber and Grassberger [1991]. Thus the method is similar although not identical to the steepest descent algorithm, although it is not at all obvious that it will always reduce noise in the general case.

3.4.3 Solution by Levenberg-Marquardt

The minimisation analogue to solving the root-finding problem by Newton-Raphson is trivial and provides no real advantages over the method described in section 3.2. However it is presented briefly since it will be used below. The Newton method for solving equation 3.4.1 is to assume that the trajectory for the minimum error lies within the linear neighbourhood of the noisy trajectory. The cost function can then be modelled with a quadratic form and a closed form solution can be written for the minimum:

$$x_{\min} = x_{cur} - \left[\frac{\partial^2 H}{\partial x^2} \right]^{-1} \nabla H \qquad \textbf{(3.4.13)}$$

Using the linearisation of the dynamical system given in equation 3.2.4 this can be written as:

$$x_{\min} = x_{cur} - \left(D^T D\right)^{-1} \nabla H \qquad \textbf{(3.4.14)}$$

where D is the matrix that was defined as before. In fact further inspection will show that this equation can be arrived at by merely pre-multiplying equation 3.2.5 by D^T and rearranging (remember $\nabla H = -D^T \epsilon$).

However equation 3.4.14 cannot easily be solved since the problem is still rank deficient. What we would like would be to achieve the stability of the gradient descent method described in the last section without sacrificing the speed of the Newton method (Newton methods converge quadratically when stable, whereas gradient descent methods only converge at a linear rate). This is in fact possible by constraining the solution to have a restricted step. Such an approach falls under the category of Levenberg-Marquardt methods (see for example Fletcher [1980]). This method provides a simple way of stabilising the Newton method while retaining the order N speed and quadratic

convergence.

Heuristically the idea is to produce an algorithm that can smoothly be changed from the gradient descent algorithm of equation 3.4.3 to the Newton method of equation 3.4.14. A closer look at these two equations shows that they are identical apart from the term in front of ∇H. Thus one obvious compromise is to use:

$$x_{new} = x_{cur} - \left(D^T D + \delta I\right)^{-1} \nabla H(x) \qquad \textbf{(3.4.15)}$$

where I is the identity matrix and δ is an arbitrary weight that defines whether the algorithm behaves more as a gradient descent or more as a Newton approach. If δ is zero then equation 3.4.15 is identical to equation 3.4.14. However when δ becomes large then the inverse becomes approximately $(1/\delta)I$, which is equivalent to equation 3.4.2. Figure 3.4.1 demonstrates the effect of this algorithm for a simple two dimensional cost function. The curve shows the set of solutions for different values of δ that map the point A towards the minimum B. For large values of δ the curve descends down the line of steepest gradient, however as $\delta \rightarrow 0$ then the solution moves more towards the minimum. In fact the theoretical arguments for the Levenberg-Marquardt procedure are more rigorous than this and it can be shown to find the minimum value for H under the constraint that the step size is restricted. Also if an appropriate method for selecting δ is used the resulting algorithm can be proved to be globally convergent and it can be shown that the convergence rate is quadratic (see Fletcher [1980] for a detailed exposition of the subject).

The most important benefit of this method is its effect on the singularities of D. Consider the singular value decomposition of $D = USV^T$. Then the matrix $M = (D^T D + \delta I)$ can be written as:

$$M = V\left(S^2 + \delta I\right)V^T \qquad \textbf{(3.4.16)}$$

Hence even though D may be singular the singular values of the M are bounded from below by δ. Indeed a further impressive feature of this algorithm is that if the non-zero singular values are bounded away from zero then, as $\delta \rightarrow 0$, the solution for x from equation 3.4.15 will tend to the solution of the rank deficient minimisation problem by the Moore-Penrose pseudo-inverse which, as we mentioned in section 3.2.4 is in some

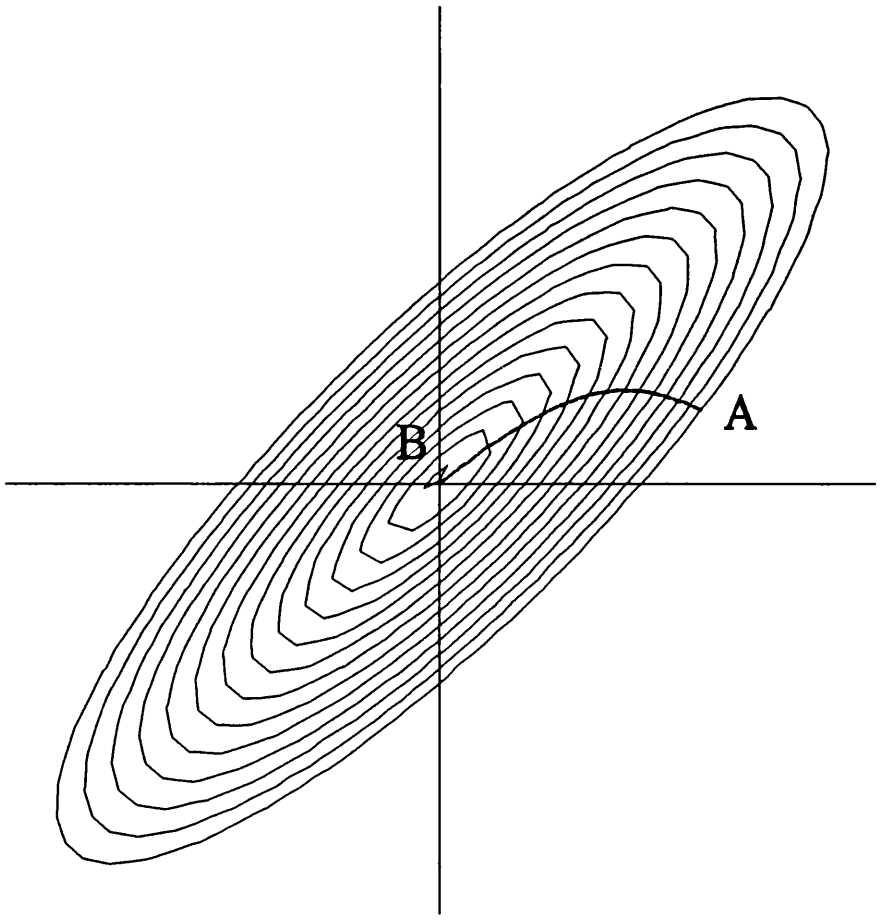


Figure 3.4.2: Levenburg-Marquardt applied to a simple two dimensional quadratic porential well. Starting at A the estimate for the minimum lies on the dotted line between A and B. If δ is small the solution is near B and is equivalent to the Newton method. If δ is large the solution is near A and is equivalent to the steepest descent solution.

sense optimal. In practice however δ cannot be allowed to become too small otherwise implementation of the algorithm will lead to numerical instabilities.

So in the directions where the singular values are identically zero this algorithm reverts completely to the steepest descent algorithm (i.e. it does not step in these directions). Also in the case of near homoclinic tangencies, where the singular values will be small but finite, although the gradient descent will dominate, this algorithm will step a small amount in the associated directions. This appears to be enough to solve the convergence problem that can occur when gradient descent is used on its own.

Finally implementation of this method is simple since M can be forced to be well-conditioned (and diagonally dominant) which allows the use of sparse matrix techniques to exploit the matrix's banded structure. Therefore equation 3.4.15 can be solved by applying Cholesky decomposition to M such that $M = GG^T$, where G is a lower triangular matrix and then solving this by applying forward and backward substitution to G and G^T respectively. All these procedures are order N and are readily found in various numerical texts, for example Golub and Van Loan [1983]. In the case of high noise levels when the linearity assumption needs to be relaxed, a larger δ can be used to restrict the step size allowable at each iteration thereby increasing the stability of the algorithm. δ can then be reduced as the data becomes cleaner.

The flexibility of the Levenberg-Marquardt algorithm makes it our preferred choice for a noise reduction procedure since the extra overheads of using this algorithm as opposed to the gradient descent method are minimal and indeed trivial if the dynamics are also being estimated (this is discussed in chapter 4), whereas the speed gains can be significant. A comparison between this method, gradient descent and manifold decomposition is given in section 3.5.

3.4.4 Other Minimisation Methods

So far we have only discussed a limited set of optimisation algorithms as methods for tackling the noise reduction problem but there are obviously other candidates that might be worth considering. Here we briefly explain our preference for the Levenberg-Marquardt and gradient descent algorithms. Our major aims in choosing one particular method over another are speed and stability. Also it must be remembered that we are also aiming to implicitly achieve shadowing. This latter aim rules out the application of any global optimisation methods such as genetic algorithms and pretty much restricts us to trying to efficiently solve the linearised problem. Two other algorithms that might seem suited to this problem are: the conjugate gradient method and a variable metric method. However these algorithms excel in solving the problem without having to calculate the full Hessian matrix. In the application here, due to the banded structure of the Hessian: D^TD, there is no real overhead in calculating this matrix and therefore these methods are unlikely to prove to be significantly better than the Levenberg-Marquardt method. This leaves the problem of stability as the deciding factor and it is this criterion that makes the Levenberg-Marquardt algorithm ideally suited to this particular minimisation problem since neither of the other two methods have the ability to restrict the size of the step taken.

3.4.5 Exact Shadowing in the Hyperbolic case

Although we have now placed the problem of noise reduction in the context of minimisation it is obvious that a shadowing orbit will be a the minimum of the cost function given in section 3.4. However this tells us nothing about any other minima that might exist. In fact if the data length is finite it is obvious that there are no minima apart from the set of deterministic orbits. This can be seen from the fact that ϵ_i is not defined for $i \leq d$ (the dimension of the state space). Thus for the slope to be zero $\epsilon_{d+1} = 0$. Then by induction if ϵ_i is zero ϵ_{i+1} must be zero. Although, in practice any time series will be finite this result relies on the lack of constraint at the ends of the time series and thus the deterministic orbit that will eventually be arrived at may well not resemble the original data at all.

A more meaningful result can be obtained by considering the case when the trajectory is of arbitrary length (the cost function will have to be divided by $\sqrt{n}$ to keep the sum bounded) and the data comes from a hyperbolic system (i.e. the attractor is hyperbolic and globally attracting). Under these conditions we can show that the only bounded minima that exist have zero error. This can be shown as follows:

If $H(z)$ is a minimum then:

$$\forall i \quad \frac{\partial H}{\partial z_i} = 0 \tag{3.4.17}$$

Hence from equation 3.4.5:

$$\forall i \quad \epsilon_i = J_i \epsilon_{i+1} \tag{3.4.18}$$

Now since the system is hyperbolic we can decompose this into stable and unstable subspaces such that:

$$\alpha_i = J_i^s \alpha_{i+1} \tag{3.4.19}$$

and

$$\beta_i = J_i^u \beta_{i+1} \tag{3.4.20}$$

now this allows us to derive a relation between β_i and β_n as $n \to \infty$:

$$\beta_i = \left(\prod_{j=i}^{n-1} J_j^u \right) \beta_n \tag{3.4.21}$$

If β_n is non-zero then β_i would be unbounded, hence the only bounded solution is for $\beta_n = 0$. Then the argument for finite data length applies and β_i must be zero for all i. A similar argument can be constructed for the errors in the stable directions.

There is of course the problem that most chaotic attractors are non-hyperbolic. In such a case the angle between the stable and unstable subspaces can become arbitrary close. In this case β_i and α_i will not necessarily be bounded even when ϵ_i is. This means that the slope can become arbitrarily close to zero (this is identical to the concept that tangencies lead to the ill-conditioning of the root finding problem).

3.5 A Worked Example

In this section we take a deterministic time series from a simple two-dimensional dynamical system (the Henon map) and add measurement noise onto it. Then it is possible to compare the effects of filtering the data with the 3 different methods described in this chapter. The details of the comparison are given below.

The time series came from the Henon map (given in chapter 2) with the usual parameter settings: $a = 1.4$, $b = 0.3$. The time series taken was the x coordinate starting at $x_1 = x_2 = 0.5$. Onto this a random *i.i.d.* signal with uniform distribution and mean zero was added. Then the data was embedded in a delay space (delay = 1). Figure 3.5.1 shows plots of the embedded data both with and without the noise added. Throughout the comparison the equations of the mapping function are assumed to be known (the alternative case is considered in detail in the next chapter) and are used to calculate the values of the derivatives for the algorithms. Each algorithm (manifold decomposition, gradient descent and Levenberg-Marquardt) is iterated 10 times and the resulting filtered time series are compared. Two measures of performance are taken. The first, dynamic error, defined as $z_i - f(z_{i-1})$, where z_i is the filtered time series, measures how deterministic the resulting trajectory is. The second, measurement error, defined as the distance between the original deterministic time series, x_i, and the filtered time series, z_i, measures how well the noise reduction algorithm has located the original data.

The manifold decomposition uses approximate manifolds, as described in section 3.2.2, with the initial stable and unstable directions set to $[1,0]$. Figure 3.5.2 shows plots of both measurement error and dynamic error, comparing the filtered time series to the unfiltered (noisy) data. It is clear from the figure that the filtered orbit is deterministic to within machine precision almost everywhere although the final orbit located is not precisely the initial deterministic orbit. If the system had been uniformly hyperbolic we would have only expected discrepancies to exist between the two "shadowing" deterministic orbits at the ends of the time series. Here we see that the presence of homoclinic tangencies means that when a shadowing orbit exists then it is not unique (the new orbit can be considered to be a shadowing orbit since the measurement error does not go above the maximum measurement error for the noisy orbit). Although the resulting

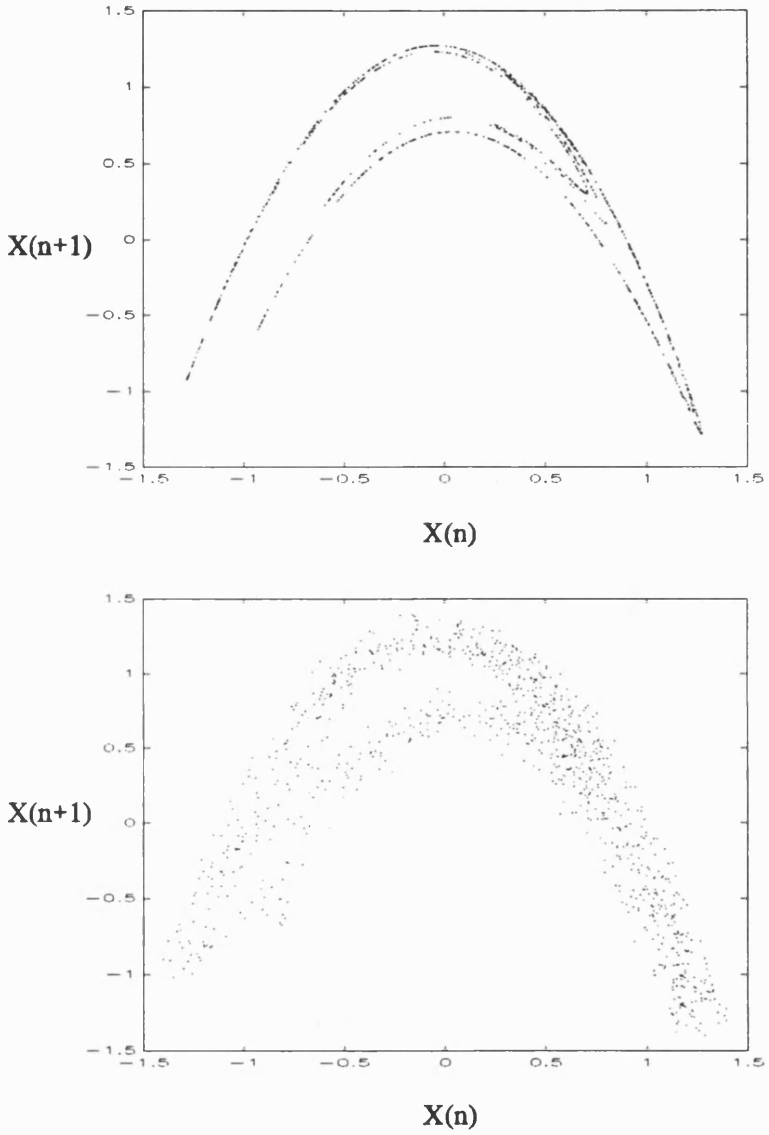


Figure 3.5.1 Plots of the delay embeddings of the deterministic signal (top) from the Henon map and then the signal after 10% noise was added (bottom)

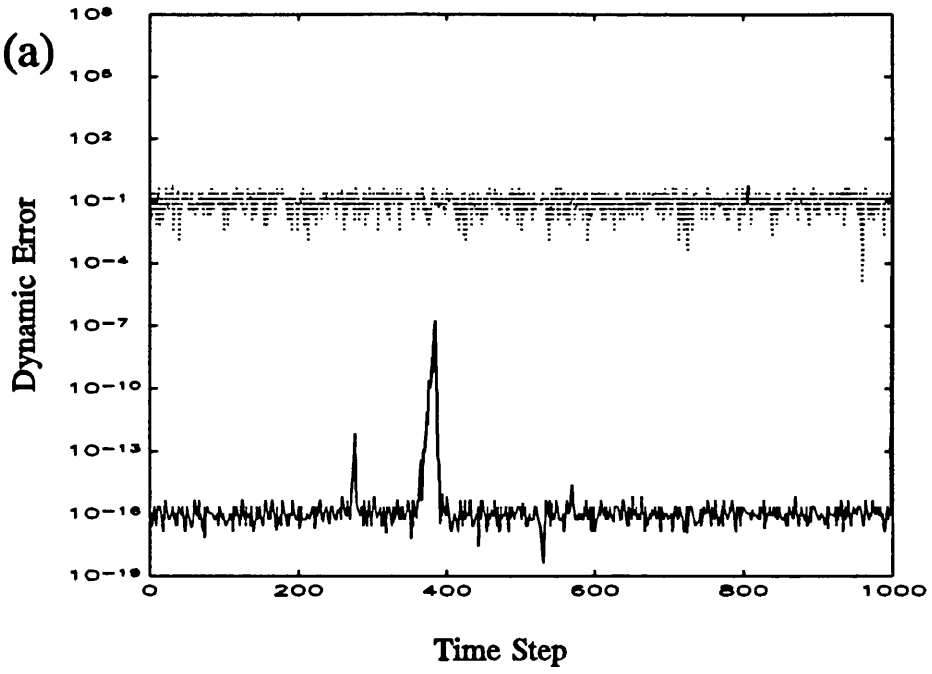


Figure 3.5.2 A comparison between the unfiltered signal (dotted) and the same signal after 10 iterates of the manifold decomposition noise reduction algorithm (continuous). (a) is a plot of dynamic error, whereas (b) is a plot of the distance from the original clean time series.

orbit is not everywhere close to the original time series, when the data is in regions of hyperbolicity the measurement error reduces down to machine precision. This is far superior to standard noise reduction method in signal processing that are based upon statistical estimates.

Figure 3.5.3 shows similar plots for the Levenberg-Marquardt algorithm. Although it is possible to implement an adaptive process to select δ at each iterate this was not used here. If only a few iterates of the algorithm are being undertaken the additional overheads of an adaptive process becomes costly (details of methods for selecting δ are described in Press *et al* [1992] and Fletcher [1980]). Here we set δ to be a constant, $\delta = 0.00001$, which has the effect of providing a guaranteed upper bound for the condition number of M in equation 3.4.16:

$$cond(M) \leq 10^5 \qquad \textbf{(3.5.1)}$$

and therefore this algorithm is likely to be more stable than the manifold decomposition. If we compare the results of this algorithm to those produced using manifold decomposition we can draw two basic conclusions. First, the regions where the dynamic error has not been reduced to machine precision are larger (although the difference between 10^{-7} and 10^{-16} is small). Thus the algorithm is slightly (but trivially) slower than manifold decomposition. Had we iterated the process further the dynamic error would probably have been reduced to the same level. Secondly, although the performance on the dynamic error is slightly worse, the measurement error is virtually indistinguishable.

In fact we can compare the two filtered time series by looking at the their absolute difference. This is plotted in figure 3.5.4. Although the difference (measurement error) between the true signal and a filtered one had many large errors due to homoclinic tangencies this is not evident in the difference between the two filtered signals does not have large errors due to tangencies. The only large differences between the two time series are at the ends and where the Levenberg-Marquardt algorithm had not converged to machine precision. This indicates that the filtered signal is practically unique ignoring the end effects. Thus the Levenberg-Marquardt method can be seen to be just as effective as manifold decomposition in ideal circumstances such as these and yet it is far easier to implement. Furthermore it is intrinsically more stable and will probably be significantly

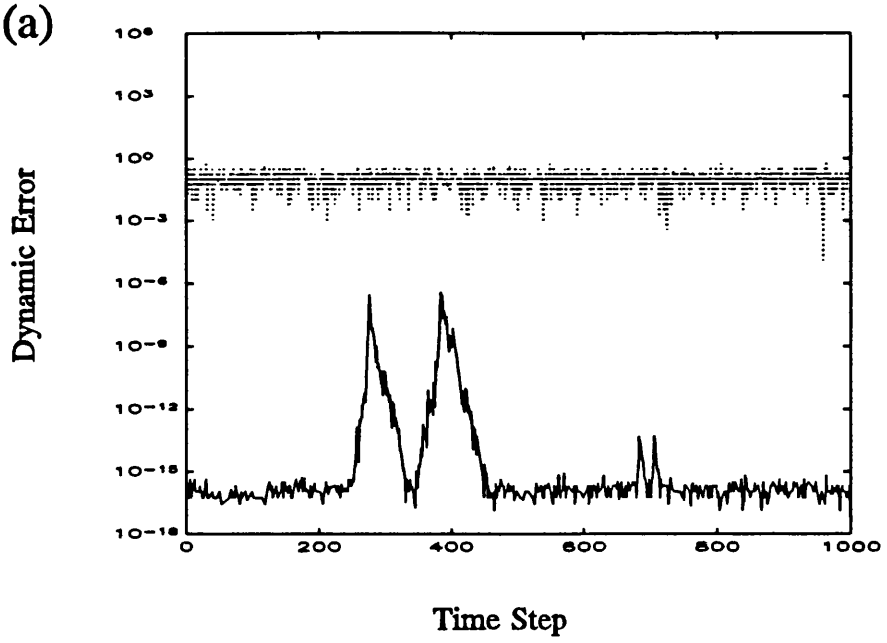


Figure 3.5.3 A comparison between the unfiltered signal (dotted) and the same signal after 10 iterates of the Levenberg-Marquardt noise reduction algorithm (continuous). (a) is a plot of dynamic error, whereas (b) is a plot of the distance from the original clean time series.

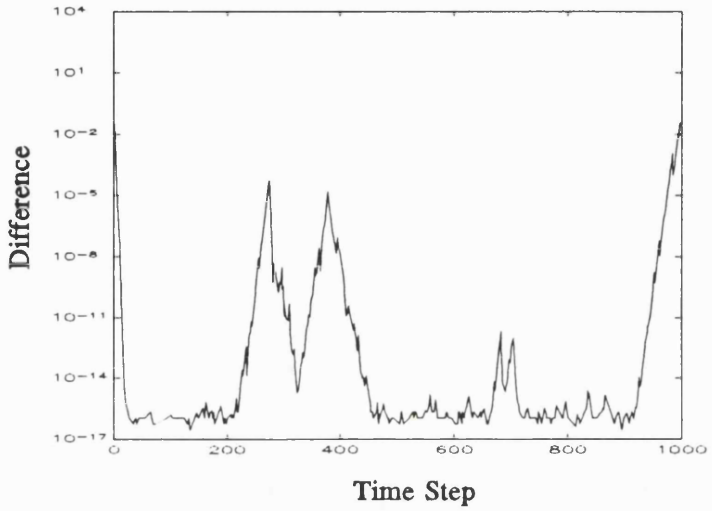


Figure 3.5.4: A plot of the absolute difference between the filtered time series using Levenberg-Marquardt and the time series using manifold decomposition.

superior for practical applications.

Finally we compare the gradient descent method to the other two algorithms. The step size chosen for the descent was 0.1. This was in no way optimised and it was selected as a reasonable value that was not unstable. The plots are shown in Figure 3.5.5. It is clear that the improvements in both deterministic error and in measurement error are very much less than were achieved with either of manifold decomposition or the Levenberg-Marquardt approaches. To a certain extent it could be argued that this is due to the speed of the gradient descent algorithm since all three algorithms were only iterated 10 times even though the gradient descent method converges at a far slower rate. However further iterations do not make significant gains over this one. This is because the algorithm tends to halt when the trajectory becomes virtually non-hyperbolic. Grassberger *et al* [1992] has observed that a non-predictive version of the gradient descent type method did not suffer from this halting problem. However in their test the algorithm had to be iterated 500 times before it could achieve a similar level of reduction in dynamic error to that shown in figures 3.5.2 and 3.5.3.

However before we write off this method we should realise that the example given here is very idealised since we have a good embedding and know the dynamics perfectly. In the following chapters we will consider more realistic scenarios where the dynamics is unknown and where a less "natural" delay space has to be chosen. It will then become apparent that the gradient descent algorithm is superior in its stability and similarly, to retain stability in the Levenberg-Marquardt algorithm, the value of δ has to be increased.

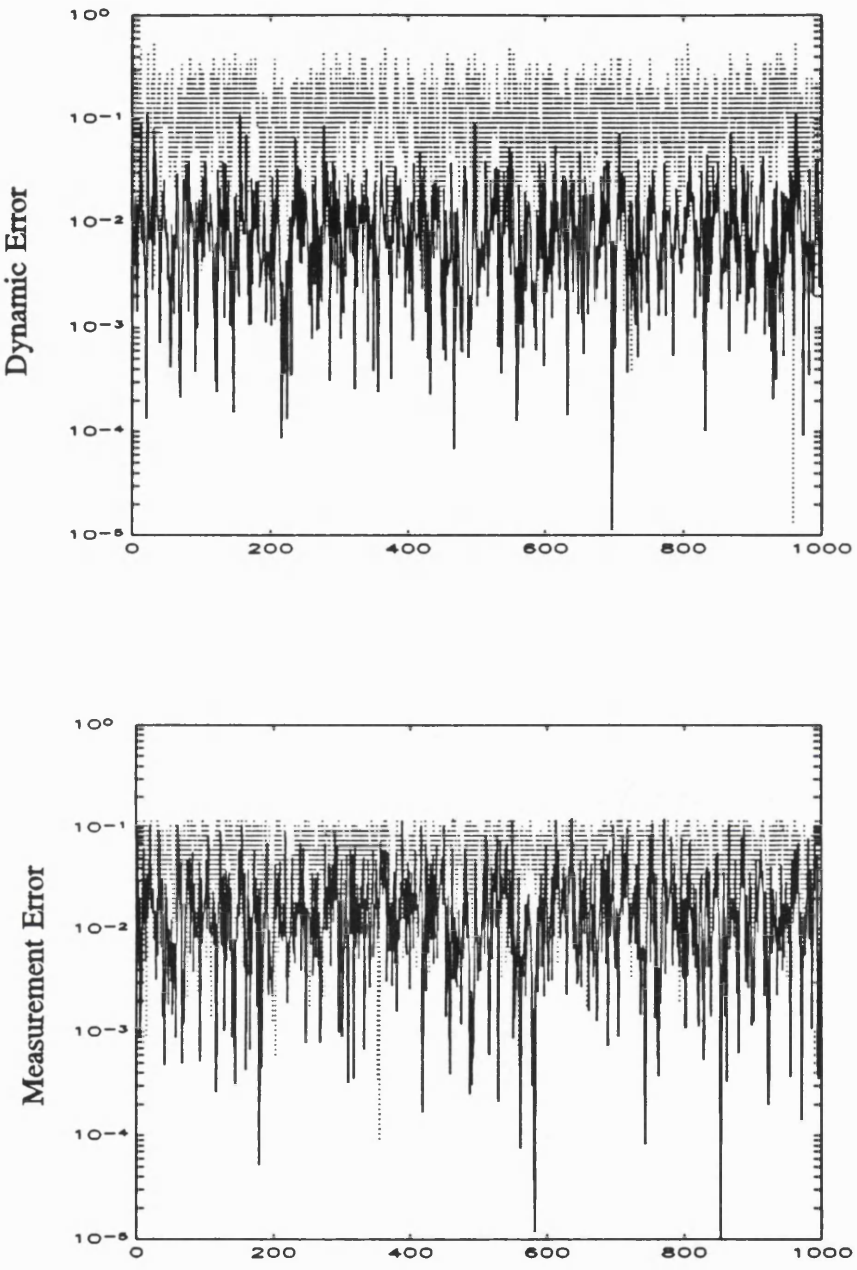


Figure 3.5.5 A comparison between the unfiltered signal (dotted) and the same signal after 10 iterates of the gradient descent noise reduction algorithm (continuous). (a) is a plot of dynamic error, whereas (b) is a plot of the distance from the original clean time series.

4. Function Approximation for Noise Reduction

In the previous chapter the problem of noise reduction was formulated as either a root finding or minimisation task. However such an approach can only really be justified when the underlying dynamical system is known *a priori*. In the case when the dynamics is unknown a method for approximating the mapping function (and its derivatives) is necessary but this makes the interpretation of noise reduction as a simple minimisation task less well defined since it is not immediately clear what the relationship between the estimation process and the noise reduction process is. Most of the algorithms that have tackled the problem of noise reduction for data from an unknown dynamical system (Farmer and Sidorowich [1991], Cawley and Hsu [1992], Schreiber and Grassberger [1991] and Sauer [1991]) have all used the *ad hoc* approach of alternating between the estimation of the dynamics and applying one iterate of their particular noise reduction scheme. This creates a variety of questions: how much noise reduction should be done between each new approximation of the dynamics; does re-fitting the dynamics improve the noise reduction at all; and is there not a more systematic approach altogether? In this chapter we investigate these problems and the relationship between estimating the dynamics and cleaning the noise towards the determinism so defined. Finally we aim to go some way to producing a more systematic approach such that the mapping function approximation can be absorbed more naturally into an extended minimisation scheme.

In section 4.1 we discuss the implications of using an inaccurate mapping function for noise reduction. This explains to some extent why the results obtained in the last chapter are difficult to reproduce when the dynamics are estimated. However, if the dynamics estimation is the bottle-neck of the process, it also opens up the possibility of further improvement by using a more appropriate method for incorporating the function estimate into the noise reduction algorithm that takes account of the relation between the two steps.

This relationship is set out in section 4.2, allowing us to conjecture what the effect of alternating the two steps would be. We then use the Levenberg-Marquardt algorithm to numerically investigate these ideas. The results agree with the general opinion that the

Newton based methods, in general cannot be applied to experimental data due to the stability problems. However we also find that even for the two step approach it is advantageous to use the Levenberg-Marquardt method to speed the algorithm up.

In section 4.2.2, we use the relationship between the two steps to propose a Levenberg-Marquardt algorithm that is extended to optimise the function approximation. This method is shown to work well even when δ is reasonably small (i.e. when the algorithm is utilising the Newton method to a large degree). This also reverses the idea that gradient descent cannot be improved upon when the dynamics are unknown.

In section 4.3 we reconsider the ideas of explicit shadowing since the arguments against it for the trajectory adjustment do not necessarily carry over to adjustments in the function parameters. An algorithm based on the method of Hammel is then outlined and this algorithm can be shown to provide a maximum likelihood estimate for the mapping function in the context of measurement noise. Unfortunately Hammel's method is still a full Newton method and this optimisation suffers the resulting instabilities. We go on to explain why it is not practical to implement a restricted step method and therefore we argue that it is preferable to use the implicit algorithm proposed in section 4.2.3.

Finally we discuss the different interpretations that can be put on the mapping estimates derived above, in comparison with the standard least squared estimate described in chapter 2. We argue that the former estimate will provide a superior model if the system from which the data came is to be viewed as purely deterministic (this is important when wishing to investigate the geometric properties of an attractor), whereas the least squared estimate will be superior for prediction.

4.1 Shadowing Nearby Maps

Any acceptable method of function approximation is required to be flexible enough to be able to provide good approximation of the dynamics being investigated while retaining enough "rigidity" to not over fit the data and merely interpolate between the points in the

data set. In practice the function approximation will not be able to exactly model the dynamics even in the absence of any noise since a finite data set requires a finite parameter set for the mapping function. This, in turn, means that it will, in general, not be possible to exactly model the dynamics. Thus there will always be a level of approximation error, as well as any measurement noise that might be present, when analyzing experimental data.

It is important to consider what effect this will have on the noise reduction process. In fact we have already touched on this problem in the last chapter when discussing Bowen's Shadowing Lemma. If we define a map $f(x_n)$ as being δ-close to the real system, $g(x_n) = x_{n+1}$ by:

$$\| f(x_n) - x_{n+1} \| \leq \delta \qquad \textbf{(4.1.1)}$$

where $\| \cdot \|$ is some norm. It is immediately obvious from the above definition that a trajectory from the real system is a δ-pseudo orbit, as defined in the Shadowing Lemma of Bowen [1970], of the mapping $f(x)$. Thus, if we assume the trajectory only lies in a hyperbolic part of the map, there exists a δ such that there is a shadowing orbit in all maps that are δ-close to the real system and remain hyperbolic along the trajectory. However, hitherto we have assumed that the noise that we are reducing is measurement noise, i.e. there always exists a shadowing orbit even if the system is non-hyperbolic, whereas this approximation noise will act as true dynamic noise since we will not be working with the true system that produced the data. Although this will have little effect when the system (or at least the pseudo-trajectory) is hyperbolic it can lead to big problems when homoclinic tangencies are present. In the worked example at the end of chapter 3 a trajectory from the Henon map was cleaned such that the remaining noise was down at the level of machine precision. This indicates that a shadowing orbit for this trajectory was accurately found, although not exactly the original orbit (there is a non-uniqueness of shadowing orbits, when they exist, for non-hyperbolic systems). If we take a deterministic orbit from the Henon map with $a = 1.4$ and $b = 0.3$ and try to clean it towards a deterministic orbit for the Henon map with $a = 1.39$ and $b = 0.3$ using the Levenberg-Marquardt algorithm ($\delta = 0.00001$) the process fails to reduce the dynamic noise to zero. Figure 4.1.1 shows the result of applying the Levenberg-Marquardt

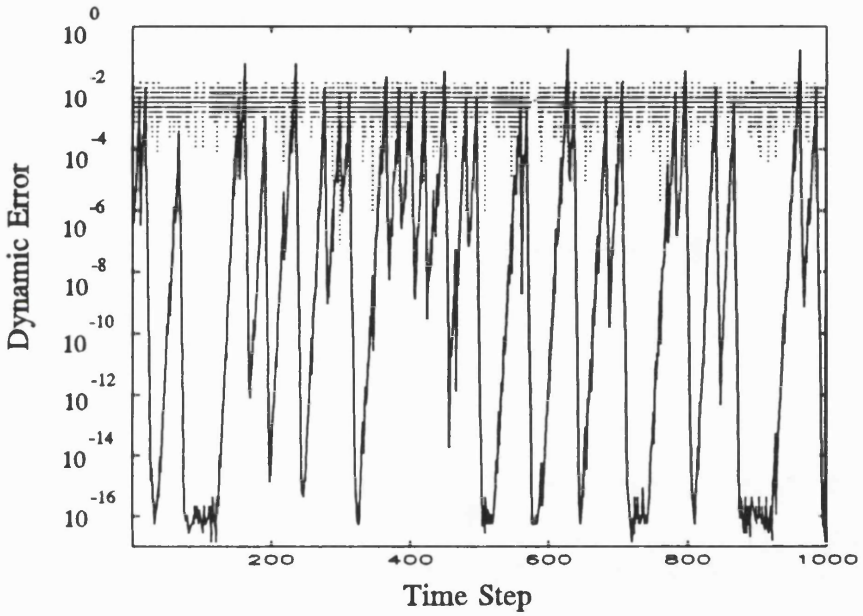


Figure 4.1.1: A plot of the dynamic error for a 1000 point time series from the Henon map (*a=1.4, b=0.3*) measured with respect to different parameters: *a=1.39, b=0.3*. The dotted line shows the original dynamic error. The continuous line shows the dynamic error after 10 iterates of the Levenberg-Marquardt noise reduction algorithm with δ = 0.00001.

algorithm 10 times to a 1000 point orbit. In section 3.5 the dynamic error was reduced to near machine precision whereas here there are points where the dynamic error has actually increased. Thus, for non-hyperbolic systems, the limitations of noise reduction associated with poor function approximation will not just be of the order of the approximation errors, as they would be with hyperbolicity. This means that function approximation plays a very important role in achieving good noise reduction and it may well be the bottle-neck of the process as a whole.

However this does not necessarily mean that if we do not know the dynamics exactly we must reconcile ourselves to being able to, at best, obtain moderate performance from our algorithms. This would only be true if there where no nearby mapping functions that contained shadowing trajectories of the orbit of interest. In fact we can hope that there will be plenty. The work by Nusse and Yorke [1988] on the shadowing of pseudo-orbits from a one-parameter family of one-dimensional maps by true orbits from nearby systems lends weight to this idea. Furthermore we are only interested in achieving this shadowing for a finite length trajectory. If there exist nearby functions that satisfy this weaker shadowing condition it is clear that some nearby maps will be far more effective than others when applying noise reduction. It would therefore be wise to try to optimise this in the noise reduction algorithm. Although may this appear to be a daunting task, in the next two sections, we will discuss some ideas that go some way towards achieving this aim.

4.2 Estimating the Dynamics

It is not immediately clear what the relationship between function approximation and the noise reduction process is. However if we consider the problem of noise reduction in terms of the minimisation of the cost function H as in chapter 3 this is the same cost function that is minimised in an Ordinary Least Squared function approximation, as described in chapter 2. The only difference between the two is that in each case the minimisation is with respect to different parameters.

Thus it is possible to consider the two processes simultaneously by using the extended cost function $H(x,p)$. That is: a function of both the trajectory and the parameter family, p, of the function approximation (although the sets of local approximations do not have explicit independent families of parameters defining them it is fair to assume their existence). This is used in section 4.2.1 to investigate how the choice of the mapping function is chosen in the two step method. In section 4.2.2 we go on to propose an optimised algorithm.

4.2.1 Alternating Noise Reduction and Dynamics Estimation

When the problem is posed in terms of the cost function $H(x,p)$, alternating between function estimation and noise reduction can be seen as minimising H by means of zigzagging down the cost function. This idea is illustrated graphically in Figure 4.2.1. However this still ignores the problem that minimising the error by making adjustments to the trajectory only implicitly tackles the shadowing problem. The cost function is still degenerate due to the existence of a $d+d_p$ dimensional minimum set, where d_p is the dimension of the parameter vector p, and thus how well the cleaned orbit shadows the original data depends on the resulting point in the minimum set that is reached. This will now be a function of how the final mapping function is chosen.

Since the shadowing problem requires that the adjustments to the trajectory are kept small the bias between reducing the error by trajectory adjustment or parameter adjustment should overwhelming tend towards optimising the mapping function estimate. This method can be justified if a gradient descent algorithm is used, since, in this case, the process takes small steps down the extended cost function, constrained to the best estimate for the mapping function. Although, theoretically, it is better to take small steps followed by repeatedly refitting the mapping function, in practice, the gradient descent method suffers on speed and the best approach will be somewhere in between a Newton type method and the gradient descent approach.

To evaluate what a good compromise might be the Levenberg-Marquardt is an ideal

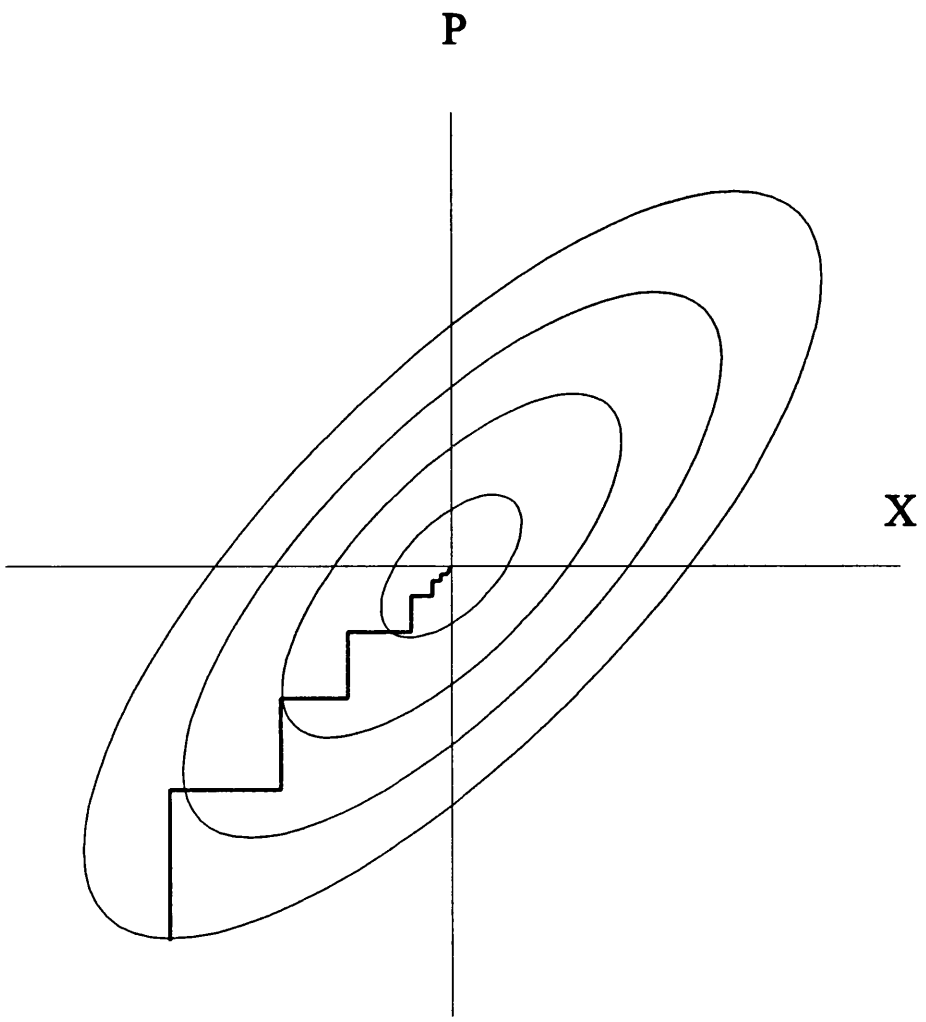


Figure 4.2.1 A graphical representation of the two step approach to noise reduction, where the adjustments to p and the adjustments to x are made independently. The resulting minimisation scheme involves zigzagging down the cost function $H(x,p)$.

algorithm since it allows us to consider the performance on a whole spectrum of methods from almost Newton to almost gradient descent by varying the Levenberg-Marquardt parameter δ. The data for this comparison was from the Henon map and was the same as that used at the end of chapter 3. The function approximation used was a set of Gaussian radial basis functions placed at the first 10 points in the embedded data set, once the data had been rescaled to span [-0.5,0.5]. The rescaling parameter in the Gaussian was set to 0.1. This function approximation was chosen since it produced a good fit, although no attempt was made to formally optimise it and the same function approximation was used in each case. Finally the embedding dimension used was 2, as before.

The mapping function was estimated using an ordinary least squared estimate and was followed by a single step of the Levenberg-Marquardt algorithm detailed in section 3.4.3. Both these stages were then repeated 10 times, after which the distance E_{dist} between the cleaned trajectory and the original deterministic orbit was measured. where E_{dist} is defined as:

$$E_{dist} = \sum_{i=1}^{n} (x - y)^2 \qquad \textbf{(4.2.1)}$$

where x is the filtered trajectory and y is the original deterministic orbit. The sum of the dynamic error was also measured:

$$E_{dyn} = \sum_{i=d+1}^{n} \epsilon_i^2 \qquad \textbf{(4.2.2)}$$

This was done for a variety of values of fixed δ until, as δ tended to zero the algorithm became unstable. The results of these tests are summarised in figure 4.2.2. It is clear from these graphs that, in practise, some benefits can be gained by using the additional information available in the quadratic form. However the algorithm loses stability as $\delta \rightarrow 0$ and a compromise is necessary to maximise the noise reduction. Without the Levenberg-Marquardt algorithm this compromise would not be possible since, even when the dynamics were known, the gradient descent method became unstable well before the step size got close to 1.0. Similarly it can been seen from the figure that a Newton type approach would also be unstable.

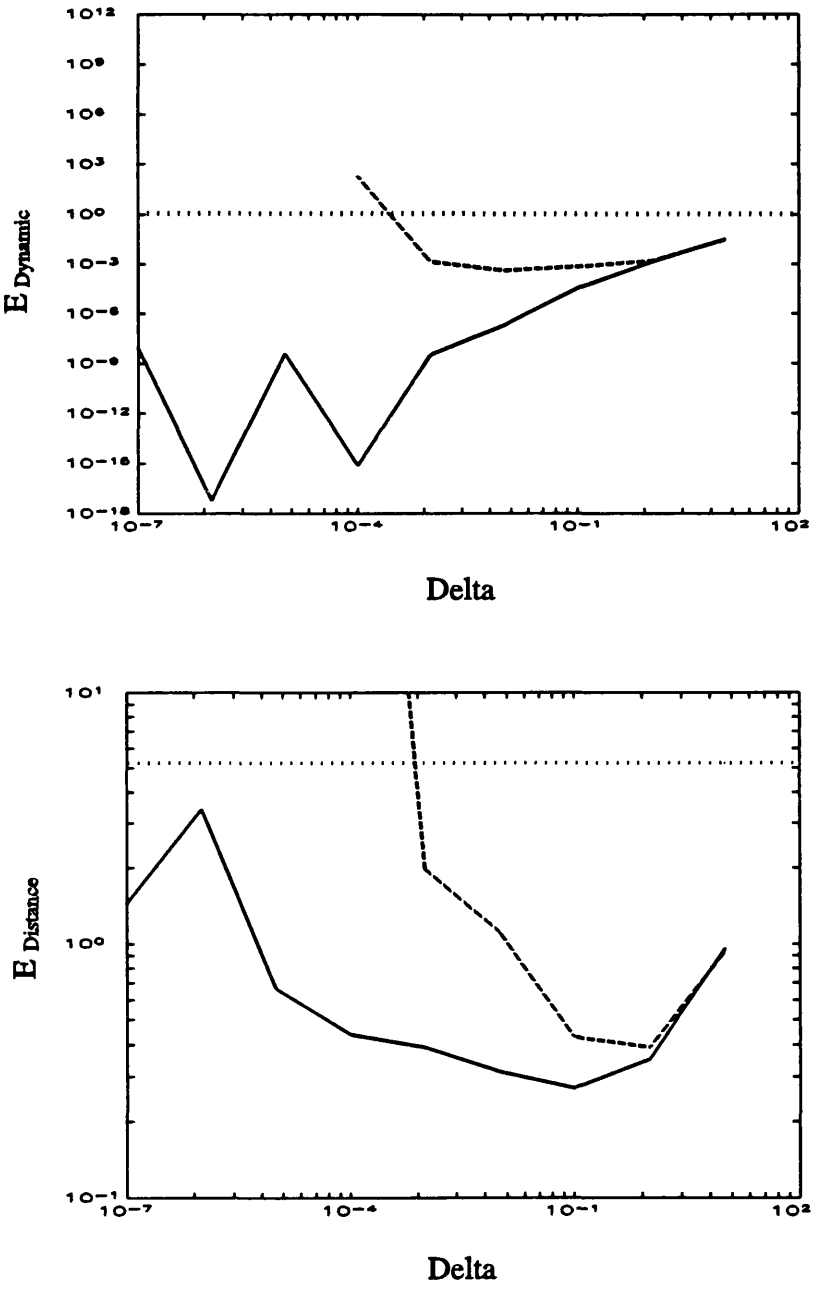


Figure 4.2.2: This picture shows the dynamic error (top) and measurement error (bottom) as functions of δ for data from the Henon map after applying the algorithms in sections 4.2.1 (dashed) and 4.2.2 (continuous). The dotted line in the both pictures is the level of the error in the unfiltered data.

Obviously this test is not conclusive since the results would probably be sensitive to the levels of noise present in the data as well as the origin of the dynamical system from which the data came. However it will serve as a good benchmark against which the algorithm in the next section can be compared. Furthermore these results are significant if the type of function approximation is local (i.e. Farmer and Sidorowich [1987]). This is because the methods described below require an explicit parameterisation of the approximation function set. Local models do not possess such a parameterisation and therefore it may not be possible to improve upon the two step method discussed here. In this case it would be sensible to either apply the Levenberg-Marquardt algorithm with a modest value of delta, or to use the gradient descent method.

4.2.2 An Extended Levenberg-Marquardt Algorithm

If we have a set of mapping functions identified by a vector p in some parameter space and a starting point in this space of maps p^* then the first order Taylor expansion of the extended error function can be written in terms of perturbations, Δp, from p^* as well as the trajectory adjustments, Δx:

$$e = \epsilon^* - D\Delta x + P\Delta p \qquad \textbf{(4.2.3)}$$

where D is the same as the definition in equation 3.2.5 and P is the matrix of partial derivatives of the mapping function around p^*, whose elements are $\partial f(x_i,p^*)/\partial p_j$ (if the mapping is a linear function with respect to p then P is the design matrix discussed in chapter 2). It is clear from this equation that the inclusion of the freedom to alter the mapping parameter p will introduce d_p additional singularities to the problem. Later we will show that these can be removed by providing extra minimisation costs to impose explicit shadowing, however initially the shadowing problem will be treated implicitly and the singularities will be removed by a Levenberg-Marquardt parameter in a similar way to section 3.4.3. For the moment let us ignore the singularities and construct the least squares cost function. This results in having to solve the following equation:

$$\begin{bmatrix} D^TD & -D^TP \\ -P^TD & P^TP \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta p \end{bmatrix} = \begin{bmatrix} D^T \\ -P^T \end{bmatrix} \begin{bmatrix} \epsilon \end{bmatrix} \qquad (4.2.4)$$

We can attempt to solve this by splitting up the problem into two simultaneous equations, the first of which is:

$$\Delta x = (D^TD)^{-1} D^T(\epsilon - P\Delta p) \qquad (4.2.5)$$

which can then be substituted into the second equation to solve for Δp:

$$P^TP\Delta p - P^TD(D^TD)^{-1}D^T(P\Delta p - \epsilon) = P^T\epsilon \qquad (4.2.6)$$

As we have already stated these equations are rank deficient and in the non-singular directions of D the above equation is trivial. Here, as before, we can solve this problem by adding an adjustment to the matrix on the left hand side in equation 4.2.4 to make it full rank and, hence, invertible.

Heuristically, and in contrast to the previous Levenberg-Marquardt adjustment, we do not wish to restrict all the free parameters since the shadowing problem only requires that the trajectory adjustment, Δx, is kept small and inflicts no constraints on the size of Δp (obviously we still need to maintain the validity of the truncation of the Taylor expansion). Therefore we only add the Levenberg-Marquardt parameter to the D^TD term by replacing it by $D^TD+\nu I$. Then equations 4.2.5 and 4.2.6 can be replaced by:

$$\Delta x = (D^TD + \nu I)^{-1} D^T(\epsilon - P\Delta p) \qquad (4.2.7)$$

and

$$P^T(I - D(D^TD + \nu I)^{-1}D^T)(P\Delta p - \epsilon) = P^T\epsilon \qquad (4.2.8)$$

which is solvable for $\nu > 0$. This is an extended version of the noise reduction algorithm

presented in section 3.4.3 that optimises both the trajectory and the mapping function.

More rigorously, we can show that this approach has a theoretical basis, in that, it provides the solution to the following constrained optimisation problem:

$$\begin{aligned} &\textit{minimise: } q(\Delta x,\Delta p) = (\epsilon - D\Delta x + P\Delta p)^T(\epsilon - D\Delta x + P\Delta p) \\ &\textit{w.r.t. } \{\Delta x,\Delta p\} \\ &\textit{subject to: } \Delta x^T\Delta x = h^2 \end{aligned}$$

where h defines the step size that Δx is restricted to. We can now prove that this is equivalent to the heuristic method described above. First we need to introduce a Lagrangian multiplier ν then we can write the Lagrangian function as:

$$\mathcal{L}(\Delta x,\Delta p,\nu) = (\epsilon - D\Delta x + P\Delta p)^T(\epsilon - D\Delta x + P\Delta p) \; . \; . \; + \nu(\Delta x^T\Delta x - h^2) \qquad \textbf{(4.2.9)}$$

A solution to the constrained optimisation can now be found by solving the $\nabla\mathcal{L} = 0$ (see Fletcher [1981]). This can be solved by considering the following equations:

$$\nabla_{\Delta x}\mathcal{L} = -2D^T(\epsilon - D\Delta x + P\Delta p) + 2\nu\Delta x = 0 \qquad \textbf{(4.2.10)}$$

$$\nabla_{\Delta p}\mathcal{L} = 2P^T(\epsilon - D\Delta x + P\Delta p) = 0 \qquad \textbf{(4.2.11)}$$

$$\nabla_{\nu}\mathcal{L} = \Delta x^T\Delta x - h^2 = 0 \qquad \textbf{(4.2.12)}$$

The first two equations, above, can be rearranged so that they are equivalent to equations 4.2.7 and 4.2.8. This merely leaves the third equation to be solved. However the freedom to choose the value of h can be substituted for the freedom to choose ν since, for every positive value of ν, there will exist a solution for Δx and Δp and consequently a value of h that solves equation 4.2.12. Thus any choice of ν can be regarded as an implicit choice of h.

In a similar way to the solution of the usual Levenberg-Marquardt, this algorithm has some useful properties. As $\nu \rightarrow 0$, this restricted step method tends to the solution for the

redundant problem (equation 4.2.4) that minimises the norm, $\|\Delta x\|$. Furthermore, as $\nu \to \infty$, the solution tends to a gradient descent method constrained to the set of minimal least squared estimates for p. This we have already stated is the two step method described in the last section. We can see this by considering equation 4.2.8, remembering that, as $\nu \to \infty$, $(D^TD+\nu I)^{-1} \sim O(1/\delta)$. This then becomes:

$$P^TP\Delta p = P^T\epsilon + O(1/\delta) \qquad \textbf{(4.2.13)}$$

which is equivalent to the least squared estimate for Δp and then Δx can be solved substituting this into equation 4.2.7. Here the term, $(\epsilon+P\Delta p)$, is the error for the new mapping function, defined by $p+\Delta p$.

The effectiveness of this method when the dynamics are unknown is demonstrated by repeating the test described in the last section. Again E_{dist} and E_{dyn} were evaluated for a range of fixed values of δ and these are given in Figure 4.2.2 for comparison with the previous results. Initially, as predicted above, the two methods appear synonymous when δ is large. Then, as δ is reduced below 0.1, the extended Levenberg-Marquardt method performs better than the *ad hoc* approach of treating the function estimation and the noise reduction separately. Furthermore this extended method can be seen to be stable for a far broader range of δ than could be achieved with the two step method.

Although neither method reduces the measurement error below about 0.3 this is misleading and should not be judged as poor performance from the noise reduction algorithms. We can see this by considering the distance between the cleaned orbit found for $\delta = 0.1$ and the unfiltered data: $E_{dist} = 5.148$, in comparison with the distance from the initial deterministic orbit and the unfiltered data: $E_{dist} = 5.257$. It is clear that the algorithm has actually found a "deterministic" orbit (although this orbit is not strictly deterministic we argue that we can ignore this since the dynamic error is only $\sim 10^{-5}$) that is *closer* to the noisy data than the initial orbit. This is therefore a statistical limitation of noise reduction since without *a priori* knowledge the unfiltered orbit would appear to be more likely to have come from the filtered data, as opposed to the original deterministic orbit. Because the limitation of the noise reduction is statistical in nature we can expect the performance to increase with the length of the data set.

However to fully compare these two approaches it is necessary to look at the speed of each algorithm. The extended Levenberg-Marquardt requires more computation and since the function approximation tends to dominate the computational expense with the manipulation of P we can see from equation 4.2.2 that the evaluation of P^TP will be quicker than the evaluation of $P^TD(D^TD+\nu I)D^TP$. However, in practice both algorithms have to perform other calculations where there is no speed difference (i.e. the calculation of the derivatives of the mapping function) and thus we do not expect the method proposed in this section to be substantially slower.

A crude comparison between the speeds of the two methods can be obtained from the user times (on the same RS6000 workstation) required to iterate each method 10 times. The two step method required 13.7 seconds of user time while the extended method required 16.98 seconds. Thus the overheads are not prohibitively increased when using this method.

4.3 Explicit Shadowing

The above approaches can be considered to be extensions of the implicit methods described in the last chapter. Here we look at optimising the mapping function parameters explicitly by minimising the distance from the original noisy data. This is analogous to the method of Farmer and Sidorowich [1991] described in section 3.3, although here we will still locate trajectory adjustments implicitly (the arguments for doing this are the same as before). A further difference between the approach taken here and that done by Farmer and Sidorowich is that they introduced Lagrangian multipliers to solve the constrained optimisation problem and then looked for a stationary point in the resulting Lagrangian function. This was done by assuming that the noise level was low and that the dynamical system could be approximated by a first order Taylor expansion. This, in turn, linearises the Lagrangian function and the approximate problem can be solved directly. Thus, using the Newton-Raphson algorithm the process can be iterated to find the stationary point.

In contrast, instead of formulating the Lagrangian and then linearising, we linearise the equations and then solve the resulting quadratic programming problem. That is: the optimisation problem has a quadratic cost function and a linear set of constraints:

$$\begin{aligned} &\textit{minimise: } q(x) = 1/2\, x^T G x + g^T x \\ &\textit{w.r.t. } x \\ &\textit{subject to: } A^T x = b \end{aligned}$$

where $x \in \boldsymbol{R}^n$, G and g^T define the quadratic cost function. The m linear constraints are then defined by an $m \times n$ matrix, A^T, and the vector b. This we solve using a *general elimination method*. This involves eliminating the number of variables by substituting the constraints into the cost function. To achieve this the variable x must be partitioned into two parts, $\{x_1, x_2\}$ such that we have:

$$q(x_1,x_2) = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} [x_1\ x_2] + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} [x_1\ x_2] \qquad \textbf{(4.3.1)}$$

where $x_1 \in \boldsymbol{R}^m$ and $x_2 \in \boldsymbol{R}^{n-m}$. We partition the constraints in a similar way and, assuming that A^T_1 is invertible, we can write:

$$x_1 = A_1^{-T} \left(b - A_2^T x_2 \right) \qquad \textbf{(4.3.2)}$$

Substituting this into $q(x_1,x_2)$ results in an unconstrained quadratic optimisation problem just in terms of x_2. This can then be solved directly. For further details about the general elimination method see Fletcher [1981].

In fact this method is solving the same linear problem that is introduced by the Newton-Raphson step in the Lagrangian function. Indeed, if this method had been employed by Farmer and Sidorowich it would have been immediately clear that their optimisation only improved the end effects. This can be seen since Hammel's method provides a solution to the linearised problem. Thus all that remains is to optimise the end constraints, x^s_1 and x^u_n.

The method therefore consists of finding a solution to the zero finding problem (here we

will use manifold decomposition) and then optimising that solution restricted to the set of possible solutions (the feasible region). Unfortunately, because Newton methods are not very stable we were unable to obtain reasonable results for a general function estimate such as radial basis functions. However, at the end of the next section, we demonstrate the idea by optimising the parameters a and b for data in the Henon map. Here it is clear that the estimated values using this method are superior to repeated least squared estimates for the parameters.

Although, in this form, this method cannot be practically applied with a general function estimate, it does provide a method that is theoretically an optimal solution in the sense of being a *maximum likelihood estimate*. We can also discuss the relationship between the implicit method described in section 4.2.2 and this method. Finally the problems of trying to stabilise this algorithm with a restricted step method are discussed.

4.3.1 An Extended Manifold Decomposition Algorithm

In this section we develop an explicit shadowing algorithm for the manifold decomposition method of Hammel. This work was originally proposed in Davies [1992]. As in the previous section, this algorithm is based around approximating the dynamics with a global parametric model, such as a combination of radial basis functions. This allows us to explicitly construct a d_p dimensional space of possible maps with which to model the dynamics. The extended zero-finding problem can be written as:

$$f(x_i+\Delta x_i\ ,\ p+\Delta p)\ -\ x_{i-1}\ -\ \Delta x_{i-1}\ =\ 0 \qquad \textbf{(4.3.3)}$$

and we wish to find the closest orbit to the original data that satisfies the above constraints. To apply the *general elimination method* we need to initially linearise the constraints:

$$D\Delta x\ -\ P\Delta p\ -\ \epsilon\ =\ 0 \qquad \textbf{(4.3.4)}$$

Since we are linearising we should aim to select our initial guess close to where we expect the optimal solution to be. The initial estimate we use for the dynamics is made

using a least squared fit to the mapping function for the reconstructed space (i.e. the cost function $H(p)$). Intuitively, minimising the approximation errors gives a good estimate for the true dynamics, although this is not a maximum likelihood method with respect to measurement errors. A more theoretical argument for using this as the initial guess for p is that the shadowing lemma relates the size of the approximation errors to the measurement errors and thus minimising the former provides a good estimate of p such that the linearisation assumptions in the algorithm are valid when minimising the latter.

To eliminate Δx from the minimisation problem we have to invert the matrix D. This we can do by using manifold decomposition (c.f. section 3.2.1). Therefore we split the problem into the stable and unstable subproblems:

$$D^s \Delta x^s - \epsilon^s - P^s \Delta p = 0 \tag{4.3.5}$$

and

$$D^u \Delta x^u - \epsilon^u - P^u \Delta p = 0 \tag{4.3.6}$$

Since the columns of P define the error ϵ as a linear function of Δp we can decompose these into stable and unstable directions in the same way that we decomposed ϵ into ϵ^s and ϵ^u. We can now use any solution to the above to approximate a point in the solution set. A simple way to obtain a solution is to fix Δp and not to consider the extended parameter space at all. Therefore we select the solution, Δx^*, as the solution to the zero-finding problem given in section 3.2.1 (i.e. the zero of the above equations with $\Delta p = 0$). We can now rewrite the problem in terms of this estimate:

$$\Delta x = \Delta x^* - D^{-1} P \Delta p \tag{4.3.7}$$

where $D^{-1}P$ is defined as the direct sum:

$$D^{-1} P := (D^s)^{-1} P^s \oplus (D^u)^{-1} P^u \tag{4.3.8}$$

where $(D^s)^{-1}P^s$ and $(D^u)^{-1}P^u$ can be solved using the algorithms proposed in equations 3.2.6 and 3.2.7 by replacing the error terms ϵ^s and ϵ^u with the columns of P^s and P^u respectively. Equation 4.3.7 hence provides a local d_p dimensional parameterisation of the zero solution set. We now wish to choose the solution that minimises the distance

from the original data, z:

$$\min \left[\sum_{i=1}^{n} (z_i - x_i - \Delta x_i)^2 \right] \qquad \textbf{(4.3.9)}$$

where x is the present estimate for the zero solution. Since the problem has already been linearised we can make a new estimate for Δp by calculating the least squared solution directly:

$$\Delta p = \left((D^{-1}P)^T (D^{-1}P) \right)^{-1} (D^{-1}P) (z - x - \Delta x^*) \qquad \textbf{(4.3.10)}$$

This is not that costly to compute since the bracketed expression that has to be inverted is a $d_p \times d_p$ matrix. Now the new estimate for Δx can be obtained by substituting the solution for Δp back into equation 4.3.7.

If the original data is assumed to be a deterministic orbit plus Gaussian noise then the maximum likelihood estimate for the mapping function is the one that is associated with the smallest (in a mean squared sense) measurement error. This is precisely what this algorithm sets out to do and, in this sense is optimal.

Although this method can be described as optimal it is not really that different from the method proposed in section 4.2.2 when $\nu \to 0$. This can be seen if we assume the linearisation of the problem to be precise. Then the problem can be solved directly by a single iterate. In such a situation both algorithms minimise the distance from the original data, ignoring the end effects. In fact the extended Levenberg-Marquardt method even minimises the distance from the original orbit with respect to the end effects.

One major difference between the two methods is in their stability. This explicit shadowing approach is still a full Newton based algorithm. Although the additional optimisation does tend to improve its stability to a certain extent (see Davies [1992]) we could not make the method stable for general function approximation (restricting the step for this method is discussed in section 4.3.2).

However we found that the algorithm was stable when used to estimate the function

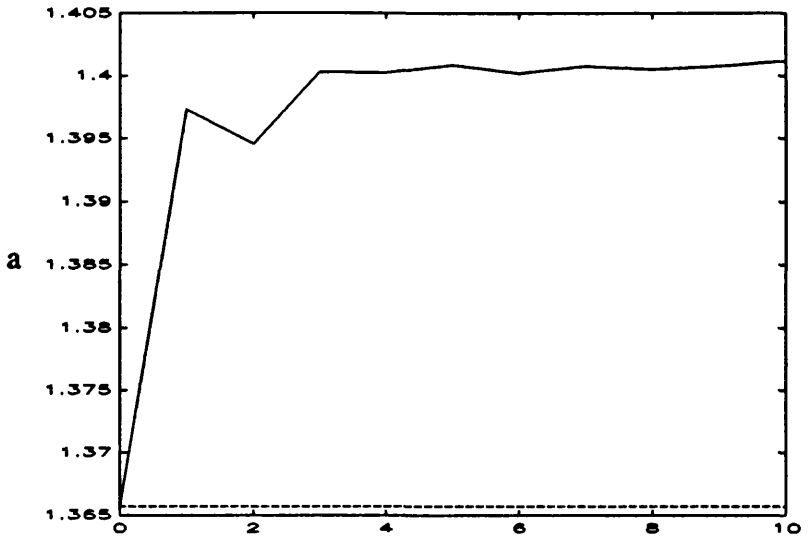
parameters a and b, of the Henon map when the basic form of the equations are known. We include a comparison of this method to the standard two step method for completeness. We applied it to the same 1000 point time series that was used in previous comparisons with $a = 1.4$ and $b = 0.3$ and the data was again embedded in a two dimensional delay space. Then the initial estimates for a and b are made using an ordinary least squared fit.

In the case of the two step algorithm manifold decomposition is applied using the estimated map. This is then followed by re-estimating the parameters by an ordinary least squared fit. The process was repeated ten times. In the explicit method proposed in this section the parameter estimates were made within the noise reduction procedure. This process was similarly repeated ten times. Figure 4.3.1 shows the estimates of the parameters a and b over the ten iterations for the method proposed here. The estimates for the two step method are not included in the plot since after 4 iterates the 2 step method had become completely unstable. This clearly demonstrates that, contrary to previous conjectures (Farmer and Sidorowich [1991]), refitting the dynamics after a Newton-based noise reduction step can actually deteriorate the estimate for the dynamics. Instead of improving the mapping, the errors in the vicinity of tangencies can distort the fit such that a new estimate is actually worse than the original.

In contrast, however, the explicit shadowing approach results in a significant improvement in the function estimates over the initial estimate. Furthermore figure 4.3.2. shows that the algorithm managed to reduce the noise level down to almost 10^{-8} and the measurement error was reduced to approximately 10^{-3}.

4.3.2 Explicit Shadowing and the Restricted Step Method

At first sight it might appear that the most natural improvement upon the last section would be to introduce into the problem a Levenberg-Marquardt parameter to enforce stability onto the algorithm. Unfortunately this is not that easy and due to the computational cost it essentially makes the algorithm impractical to implement.



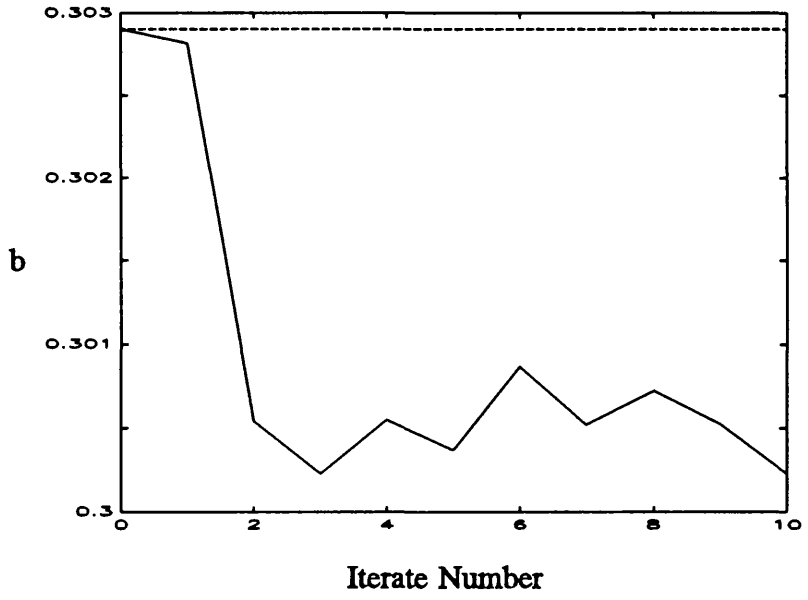


Figure 4.3.1: A comparison between the ordinary least squared estimate (dotted) and the optimised estimate (continuous) for the parameters *a* and *b* in the Henon map.

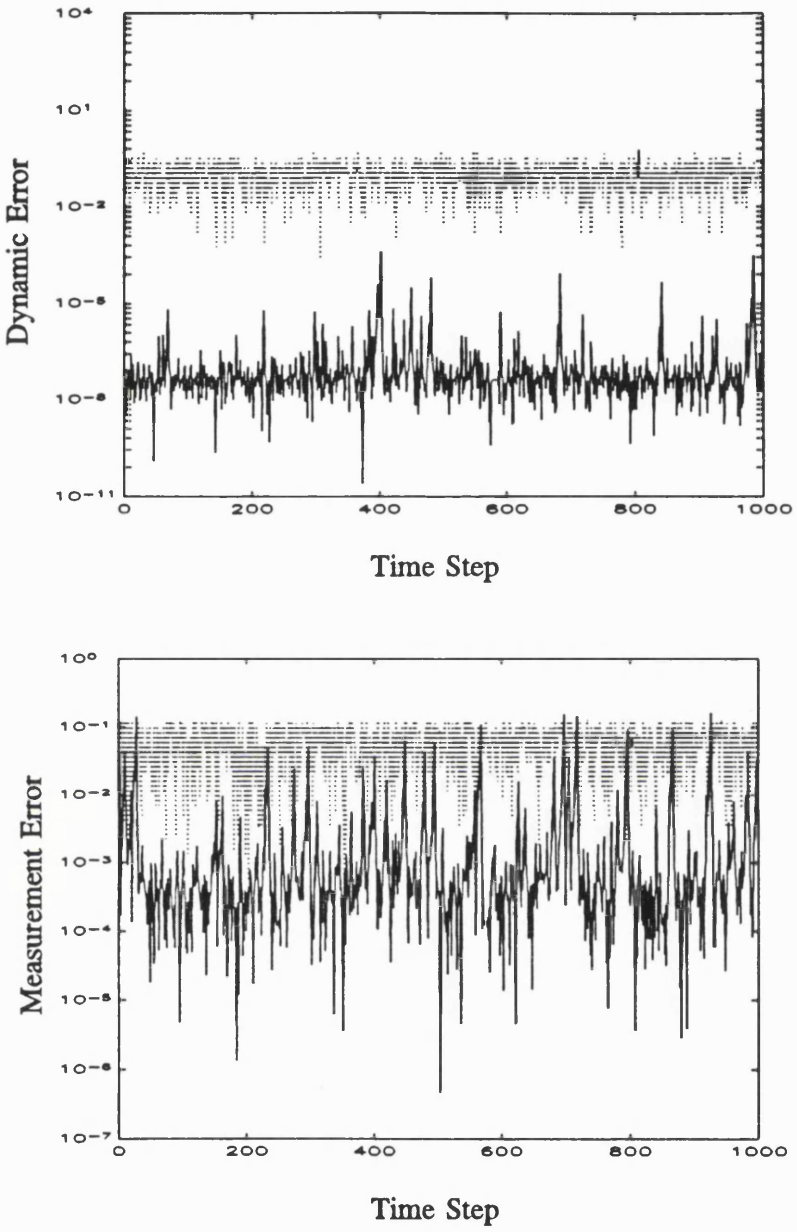


Figure 4.3.2: A comparison between the noisy signal (dotted) and the same signal after 10 iterates of the extended manifold decomposition algorithm (continuous). The top picture shows the dynamic error, whereas the bottom picture shows the measurement error.

The main problem with attempting to apply a restricted step method to a constrained optimisation task is that the standard approaches to solving such problems do not transform it into a simple minimisation scheme. Instead the solution is a stationary point of the associated Lagrangian function. Thus the solution could be a saddle instead of a minimum. In this case, the steepest descent direction will not step towards the solution. Indeed Levenberg-Marquardt adjustments are often used, in the vicinity of saddles, specifically to *stop* Newton-based algorithms from finding the saddle instead of the required minimum (see Fletcher [1980]).

We can still consider transforming the saddle into a minimum. This is easily done by decomposing the Hessian matrix into its corresponding eigenvectors and eigenvalues. Then reversing the negative eigenvalues (the eigenvalues are necessarily real since the Hessian matrix is the square of matrix with real elements) will result in a positive semi-definite matrix. This we can guarantee has a minimum and it is then a simple task to incorporate a restricted step into the solution. However, eigen-decomposition, like SVD, cannot take advantage of the banded structure of the matrix. Therefore the speed of the algorithm would require in the order of N^3 operations. In practice this is too costly to implement.

It may be possible to develop some other form of restricted step method, or at least one that forces the problem to be well conditioned. However we have not pursued this direction here. Instead we favoured using the implicit algorithm proposed in section 4.2.3. Although this does not explicitly minimise the distance between the filtered trajectory and the original data, it does minimise the adjustment at each linearised step. It also is easy to implement and can be adjusted from a gradient descent approach to a Newton approach with ease.

4.4 Improved Deterministic Modelling

If we are to incorporate the mapping function during the noise reduction process whether by the two step method or one of the other methods proposed here, we should consider what the difference is between the estimate by an ordinary least squared fit and, for example, the optimal estimate proposed in section 4.3. The difference between them is in the assumptions that are implicit in the methods. An ordinary least squared estimate assumes that the embedded data is exact and that any inaccuracies are due to dynamic error. This type of estimate is best suited to one-step prediction.

In contrast, the optimal method described in section 4.3 assumes that the dynamic error is zero and that any errors present are due to measurement noise. This estimate provides the best model for a system that possesses an orbit close to the original data. Therefore, if we believe that the behaviour of the series can be solely attributed to a deterministic dynamical system the best model for the dynamics is to assume that the inaccuracies are measurement noise. For example if we wished to investigate the geometric properties of the attractor from which the data came, we need our model to be able to exhibit all the long term behaviour that the time series does. This is not guaranteed if an ordinary least squared estimate is used.

In reality there will be both measurement noise and dynamic noise and the statistical problem becomes less well defined. However if we wish to ignore the dynamic noise to allow us to study a deterministic system (this assumes some form of shadowing) we still require a mapping function that models the system's long term behaviour. This is precisely what we achieve if we can reduce the dynamic noise to zero and, heuristically, we argue that a mapping function estimate that improves the ability to reduce the dynamic noise by trajectory adjustment is more likely to model the long term behaviour observed in the time series. This argument is necessary since we will never actually be able to reduce the dynamic noise to zero. An application of this idea to some experimental laser data is presented in chapter 6.

5. Noise Reduction: Numerical Properties

This chapter discusses some of the properties that we can expect from the noise reduction algorithms described previously. We will concentrate on the Levenberg-Marquardt procedures (using the algorithm from section 3.4.4 when the dynamics are known and the algorithm from section 4.2.2 otherwise) since this encompasses most of the basic characteristics of the other methods.

In section 5.1 we consider the end effects on the measurement error resulting from filtering the data when the dynamics are known. These effects also govern the error propagation in the vicinity of homoclinic tangencies. From the linearised equations in chapter 3 we predict the rate of growth or decay of the errors at the ends as a function of the Lyapunov exponents of the underlying dynamical system. We then go on to compare the actual scaling in a numerical example to our prediction and we show that they are in general agreement. Finally we note that, although these effects are prominent when the dynamics are known, if we are approximating the dynamics then the associated errors of the approximation will tend to swamp the predicted scaling.

Section 5.2 contains a numerical investigation of the rate at which the noise reduction algorithm converges towards a deterministic orbit. We would expect the algorithm to converge at a linear rate for large δ and at a quadratic rate for small δ unless the presence of tangencies reduces the performance. This is investigated for a variety of values of δ and the rate of convergence is seen to agree with the theory for small noise levels. However at higher noise levels the ability to attain quadratic convergence is reduced.

Finally, in section 5.3, we look at the performance of the noise reduction algorithm as a function of the noise level. Although it is not surprising that the performance decreases as the noise level is increased we find that the same occurs at very small noise levels. This is due to the approximation errors inherent in the modelling procedure which swamp the added noise for small noise levels.

5.1 End Effects and Error Propagation

The noise reduction algorithms described in chapter 3 rely on using information from upstream and downstream of a given point to reposition that point. Thus at the ends of the time series where there is less data to use we would expect the noise reduction algorithm's performance to be worse. We can see this by considering the linearised problem set out in section 3.2 and its solution by manifold decomposition (although the resulting scaling laws apply to any Newton based solution). This reduces the set of possible solutions to initial guesses for ϵ^s_1 and ϵ^u_n. Then the error between the filtered time series and the original clean time series (assuming the linearisation assumption to be accurate) will depend on the difference between the actual values for ϵ^s_1 and ϵ^u_n and the assumed values (usually zero). The effects of these differences will then propagate up or down the time series. As we have already discussed these errors should decrease when propagated along the time series as long as the manifold decomposition used is 'typical'. In fact, if the linearisation is accurate we can write an expression for the propagated error:

$$\begin{aligned} \epsilon^s_1 &\neq 0 \\ \epsilon^s_i &= \prod_{j=1}^{i-1} J^s_j \cdot \epsilon^s_1 \end{aligned} \qquad \textbf{(5.1.1)}$$

similarly if we consider the last point in the time series it is obvious that:

$$\begin{aligned} \epsilon^u_n &\neq 0 \\ \epsilon^u_{n-i} &= \prod_{j=n-1}^{n-i} (J^u_j)^{-1} \cdot \epsilon^u_n \end{aligned} \qquad \textbf{(5.1.2)}$$

Furthermore if we assume that the system that produced the original time series is ergodic then the product terms in the above expressions can be approximated as functions of the Lyapunov exponents associated with the attractor:

$$\lim_{n\to\infty} \; n \log \left[\prod_{i=1}^{n} J^s_i \right] \; \rightarrow \; \lambda_- \qquad \textbf{(5.1.3)}$$

and

$$\lim_{n \to -\infty} \; n \log \left[\prod_{i=1}^{n} J_i^u \right] \; \rightarrow \; -\lambda_- \qquad (5.1.4)$$

where λ_- is the *largest* negative Lyapunov exponent (i.e. the one closest to zero) and λ_+ is the *smallest* positive Lyapunov exponent. Thus we can expect the propagated error to approximately reduce at an exponential rate as a function of the relevant Lyapunov exponents as follows:

$$\epsilon_k \approx e^{k\lambda_-} \epsilon_1 \qquad (5.1.5)$$

and

$$\epsilon_{n-k} \approx e^{-k\lambda_-} \epsilon_n \qquad (5.1.6)$$

It is important to note that the critical Lyapunov exponents are those that are closest to zero. This immediately implies that applying any noise reduction algorithm to a flow will be limited by the flow direction.

The scaling of errors at the ends of the time series it is equally applicable to points anywhere along the signal if the dynamics are not hyperbolic. This is where the linearisation breaks down and hence inaccuracies are introduced to the filtering process. Then the propagation of these errors will tend to scale in the same manner as described above. The result of this is that the measurement error after filtering has a jagged zigzag appearance.

As an example of this scaling a 500 point data set from the Henon map, with 10% noise added, was filtered with the Levenberg-Marquardt procedure ($\delta = 0.00001$) using the known dynamics. For this time series the Lyapunov exponents were computed to be:

$$\lambda_+ = 0.397$$
$$\lambda_- = -1.596$$

using the QR algorithm described in chapter 1 (see also Eckmann and Ruelle [1985]). These values should then approximate the positive and negative slopes in the absolute value of the measurement error when plotted on a *log* scale. This is shown in Figure 5.1.1. In the blown up section the solid line is the measurement error and the dotted line

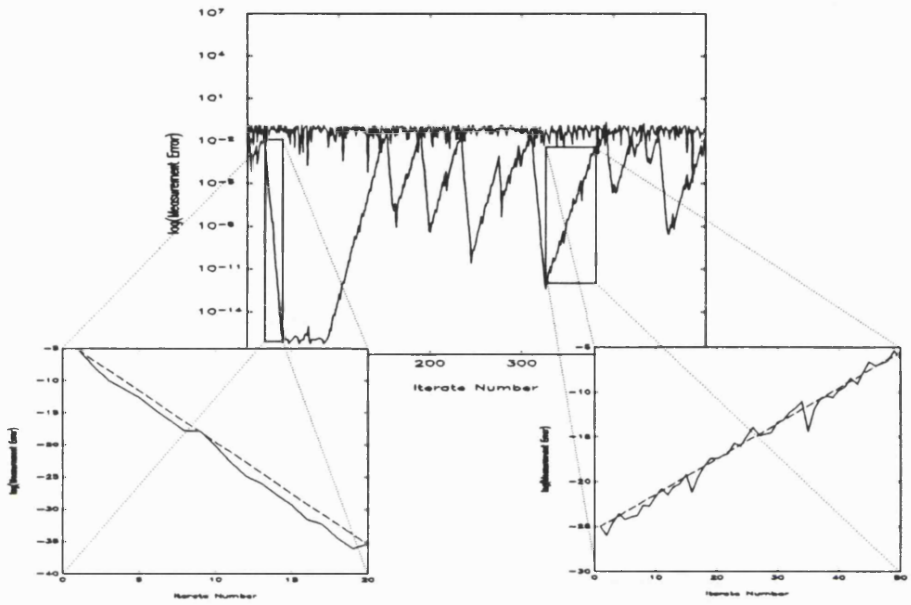


Figure 5.1.1: The top plot shows the measurement error for a 500 point time series from the Henon map after noise reduction. The bottom plots compare the negative (left) and positive (right) slopes in the measurement error with the predicted gradient using the Lyapunov exponents.

is the gradient of the relevant Lyapunov exponent. Clearly the scaling laws accurately predict the error propagation along the time series.

So far we have only observed the scaling when the known dynamics have been used in the noise reduction process. unfortunately such scaling does not appear to persist when the dynamics have to be approximated. This is not really surprising since, as we noted in chapter 4, the ability to reduce the measurement error is severely restricted when approximating the dynamics and therefore there is not such a large range in the measurement error over which to identify the Lyapunov scaling. This means the statistical noise due to the approximation of the mapping function swamps any scaling, which is itself only a statistical observation. Even when the correct dynamic model is being used and only the parameters are being fitted (cf. section 4.3.1) it is questionable whether any scaling due to the end effects and tangencies is discernable (see figure 4.3.2).

5.2 Convergence to Zero Dynamic Error

In chapter 3 we introduced the Levenberg-Marquardt algorithm to try to preserve the stability of the gradient descent approach while improving the convergence rate of the algorithm. The Levenberg-Marquardt parameter, δ, should provide a spectrum of algorithms with convergence rates spanning from linear to quadratic. However as δ is reduced the stability of the algorithm becomes less certain. Thus we need to investigate to what extent we can achieve quadratic convergence. Assuming that we are close to the minimum, the theory states that, for a fixed value of δ, in the well conditioned directions the convergence will be virtually quadratic, and in the badly conditioned directions the convergence will be approximately linear. Therefore we should expect that initially the overall errors will reduce at a quadratic rate until the errors in the well conditioned directions are negligible. Then the errors in the badly conditioned directions will dominate in the process and the convergence will be no better than linear. Also the choice of δ should govern how soon the convergence rate ceases to be quadratic, at least until the linearisation assumption breaks down.

To examine the performance in practice we initially took a 1000 point time series from

the Henon map and added 1% additive noise. We then used the Levenberg-Marquardt algorithm proposed in section 4.2.2, along with the same radial basis function approximation used in that section. Then for a variety of fixed values of δ we determined the dynamic error after each iterate of the noise reduction procedure. The results are presented in figure 5.2.1. The figure shows the graph of the square root of the dynamic error for the $k+1$th iterate of the algorithm against the square root of the dynamic error for the kth iterate, plotted on a *log-log* scale. This makes sense since the convergence relationship is expected to be:

$$E_{k+1}^{1/2} \propto \left(E_k^{1/2} \right)^q \qquad \textbf{(5.2.1)}$$

where E is the dynamic error and q is the rate of convergence. Thus the gradient on the plot in figure 5.2.1 indicates the rate of convergence q. It is not surprising that for the larger values of δ the algorithm becomes increasingly stable and the measured rate of convergence is approximately the expected linear ($q = 1$) rate. At smaller values of δ, the graphs in figure 5.2.1 can, in general, be seen as having two distinct gradients. The initial slope of the graph for $\delta < 10$ is greater than one and at $\delta = 0.01$ is approximately two. In all cases the gradient of the lower sections of the curves is practically one. Finally as δ is decreased below 0.01 to 0.001 the two stage structure of the curve is destroyed and initially the convergence rate is less than two, although when $E^{1/2}_n \approx 10^{-3}$ the rate does appear to increase. Then at $E^{1/2}_n \approx 10^{-4}$ the convergence rate becomes linear. Thus for small noise levels, until linearisation assumptions break down due to ill-conditioning, the convergence rates agree in general with those predicted for the algorithm.

If we then repeat the test with an added noise level of 10% we find that the convergence rates do not perform as well although, as before, for each value of δ the algorithm still eventually converges at a linear rate. This is shown in figure 5.2.2. Here the two stage structure appears to break down when δ is less than 0.1 and at $\delta = 0.001$ there is little evidence of any quadratic convergence. Thus the Levenberg-Marquardt algorithm does allow us to achieve an improvement in the convergence rate over the gradient descent approach. However, as the level of noise is increased the improvements obtainable are reduced.

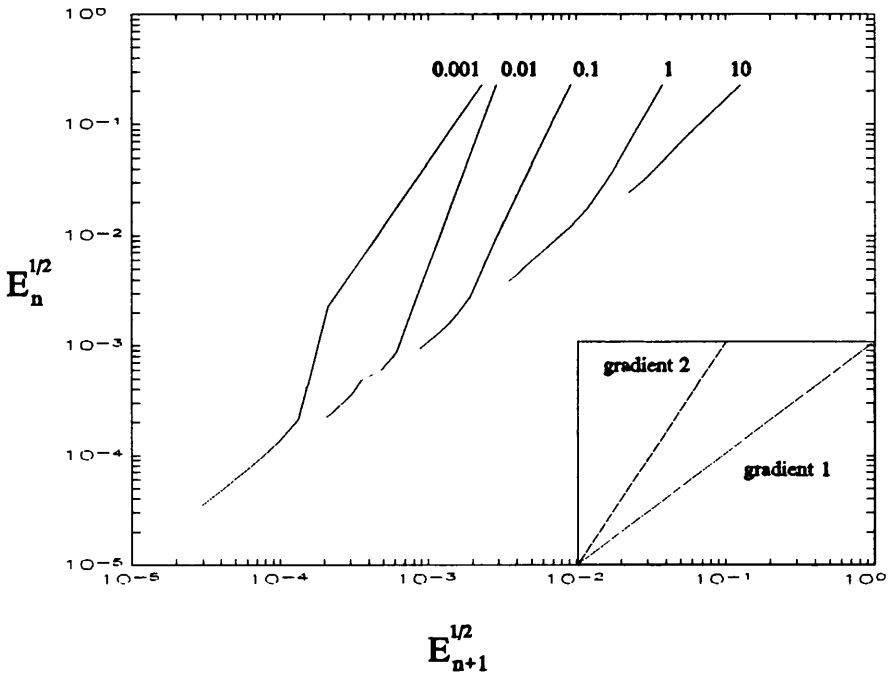


Figure 5.2.1: A plot of the root of the dynamic error against the value for the previous iterate. The value of delta for each curve is indicated on the plot. The initial noise level was set at 1%.

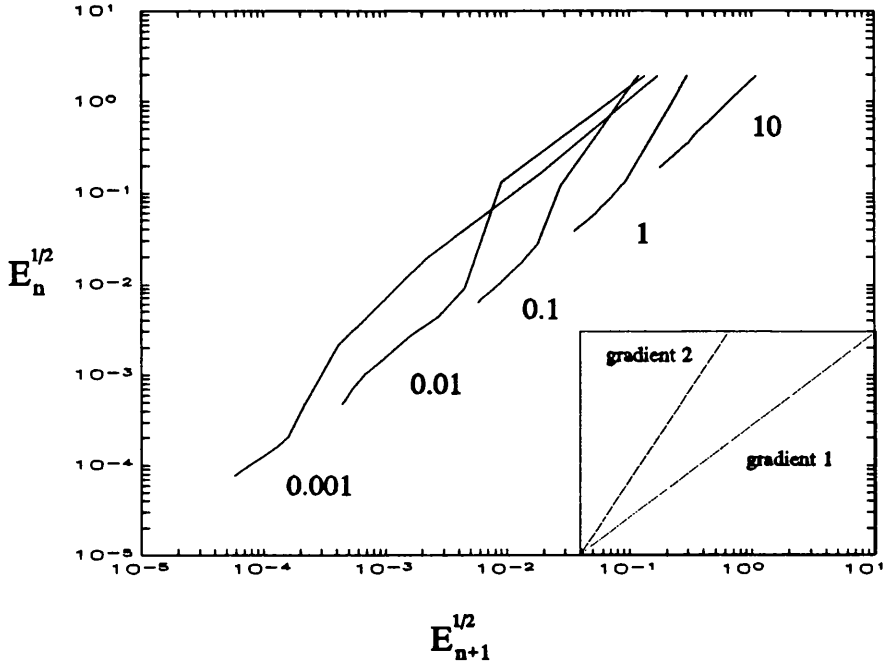


Figure 5.2.2: A plot of the root of the dynamic error against the value for the previous iterate. The value of delta for each curve is indicated on the plot. The initial noise level was set at 10%.

5.3 Improvements in Signal-to-Noise Ratio

As we have already seen in section 5.1, approximating the dynamics appears to limit the performance of the noise reduction algorithm in terms of measurement error. Here we look at how this limitation is effected by the level of the added noise and as before we can expect the function approximation to be the limiting factor.

Initially we took a 1000 point time series from the Henon map corrupted with various amounts of uniformly distributed additive noise. The noise reduction algorithm used was the Levenberg-Marquardt algorithm ($\delta = 0.1$) described in section 4.2.2 and the dynamics was modelled in exactly the same way with 10 radial basis functions. The performance of the algorithm is measured as the percentage increase in signal-to-noise ratio, θ:

$$r_k = \frac{E^0_{dist}}{E^k_{dist}} \qquad \textbf{(5.3.1)}$$

where E^k_{dist} is the measurement error after k iterates of the noise reduction algorithm (here, as before we will use $k = 10$). The results are shown in figure 5.3.1. From this graph it is clear that there is an optimal level of noise at which the noise reduction algorithm performs best. This is because there are two major limitations in the function approximation. First the need to choose a restricted set of functions with which to model the dynamics automatically means that even in the case of no added noise there will be a residual level of approximation errors. Thus the noise reduction algorithm will clean the data towards a deterministic orbit in the approximated map and if the noise is dominated by the approximation errors the performance, r, will be less than one. In our test the performance was less than one for 0.1% noise. Then as the noise level was increased the performance rose rapidly to a maximum at around 10%. At higher levels of noise the algorithm will be limited by the statistical uncertainties in the fitting procedure and thus we expect the performance to monotonically decrease as more noise is added. The graph in figure 5.3.1 confirms this to be the case.

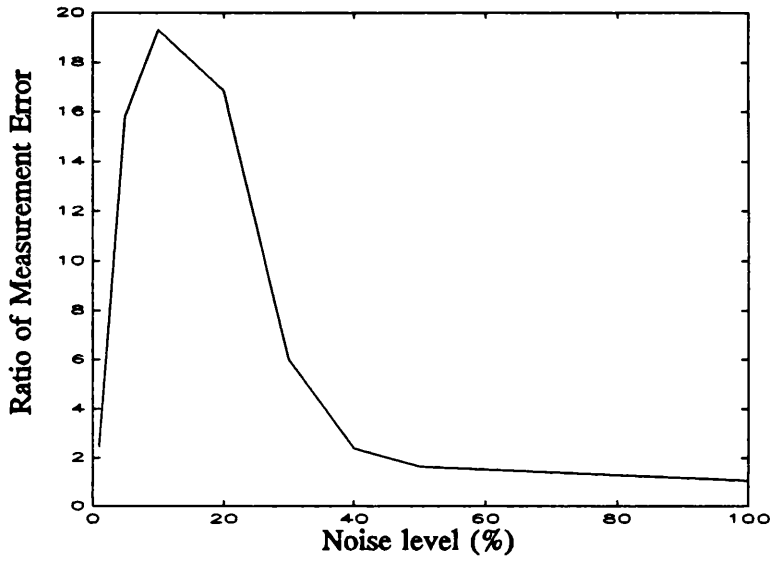


Figure 5.3.1: This plot shows the performance of the noise reduction algorithm after 10 iterates, as a function of the initial noise level.

6. Noise Reduction: Applications.

We are now in a position to apply our noise reduction techniques to some more general problems. In this chapter we show that this kind of noise reduction can equally be applied to data that is embedded in a state space using more sophisticated embedding techniques than delays. The particular method that is used here is the Singular Systems approach discussed in chapter 2, although we could have used any linearly filtered (FIR) delay space.

In section 6.1 we apply the noise reduction algorithm proposed in section 4.2.2 to data sets from two different attractors from the Lorenz system. The first is the classic chaotic attractor that occurs in the Lorenz equations at the parameter values: r = 28, b = 8/3 and σ = 10. The second is a periodic orbit that occurs at the parameter values: r = 100.5, b = 8/3 and σ = 10. These two attractors allow us to assess the performance of the noise reduction algorithm in cleaning data that was produced by a flow as opposed to a map. Furthermore we show that this method is just as applicable to a non-chaotic time series as it is to a chaotic one. Also, since these time series are significantly over-sampled a traditional low-pass filter should perform well at reducing the noise. Thus we compare the performance of our algorithm to that of a standard low-pass filter.

We then go on to consider reversing the roles of the dynamics and the noise. That is we address the problem of removing an unwanted dynamical signal from an unknown "non-deterministic" one. This is, in fact, the problem that noise reduction already solves since we can think of the algorithm as separating two signals using low dimensional determinism as the discriminator. Once this is done it is our choice which signal we choose to keep and which to throw away.

Finally in our last application we demonstrate how a deterministic model for a time series can be improved by applying noise reduction. The basic idea has already been discussed in section 4.4. The aim is to find a mapping function that has an orbit that is close to the original orbit and has no dynamic noise. In practice, we merely reduce the dynamic noise as much as possible. We then assume that the map associated with the filtered data is more likely to possess the correct dynamical behaviour of the true system. To

demonstrate the effectiveness of this method we apply it to some experimental laser data.

6.1 Noise Reduction Applied to the Lorenz System

The Lorenz model for convection provides a good example of a low dimensional nonlinear vector field that can exhibit chaos. There are two important reasons for wanting to test our noise reduction method on a time series that has come from a flow. First of all, a time τ map from a flow cannot, in general, be expressed as an analytic function. This provides added difficulties in modelling the dynamics since most global function models seem to be able to approximate analytic functions better than non-analytic ones.

The second challenge is that, if the data is sampled at a fairly high rate, a simple delay state space will not necessarily produce a good embedding of the data in the presence of noise. Thus we have to adapt the noise reduction algorithm to cope with a more sophisticated embedding procedure. Here we will use singular systems analysis to choose our embedding space. However the alterations required to work in such a space are minor and apply to any reconstructed space produced from a set of linear FIR filters .

We have to work with vectors $Y_i = L\,X_i$, where L is an $p \times q$ matrix and X_i is the q-dimensional delay window: $\{x_i, x_{i-1}, \ldots, x_{i-q+1}\}$ and Y_i is the p-dimensional state space vector for X_i. We can now define the map that we intend to approximate:

$$x_{i+1} = f(Y_i) = f(LX_i) \qquad \textbf{(6.1.1)}$$

The only other change required to the algorithm is to adjust the derivatives of the map with respect to the data points, x_i.

The algorithm used below was the extended Levenberg-Marquardt method proposed in section 4.2.2 with the Levenberg-Marquardt parameter kept fixed at 0.1. This does not allow us to completely eliminate the noise, but it does ensure stability. In both cases the data was initially rescaled to span [-0.5,0.5] and the delay window was chosen to be 10 data points in length. The embedding space was then constructed using the first 3 singular directions. The dynamics were approximated with 100 Gaussian radial basis functions

with the centres chosen from the data set and the rescaling parameter set at 1.0. Finally the algorithm was applied ten times to the data sets in each case.

6.1.1 Data from a Chaotic Attractor

For our first data set we used the standard chaotic attractor that occurs in the Lorenz equations at parameter values: $r = 28$, $b = 8/3$ and $\sigma = 10$. The equations were integrated with a fixed step size Runge-Kutta algorithm with a step size of 0.01. Initially the equations were integrated for 1000 steps to allow the trajectory to settle down onto the attractor. We then took the next 5000 iterates as our data set. The time history of the data is shown in the top picture in figure 6.1.1. The bottom picture shows the state space reconstruction with the first two singular directions (c.f. section 2.3.3).

Figure 6.1.2 shows the same data with some noise added. The noise was I.I.D. with a uniform probability distribution. The noise level was set so that the total range of the noise was 10% of the total range of the data. It is clear from figure 6.1.2 that it is no longer possible to identify adjacent points in time from the embedded data. However the time history shows that there is still a lot of the data's structure visible in the noisy time series. This is because the sample rate for the data was quite high (we will discuss the ability to clean the time series with a low-pass filter below).

We then applied our noise reduction algorithm to this noisy time series. Figure 6.1.3 shows the results. The initial dynamic error prior to applying noise reduction was 4.88 and after noise reduction it had been reduced to 2.18×10^{-4}. This can be seen in the time history plot where the data looks very similar to the original deterministic trajectory. Also looking at the state space (bottom) we can see that it is now possible to identify adjacent points in time from the embedded data (i.e. determine the direction of the flow). The one noticeable difference between the embedded data in figures 6.1.1 and 6.1.3 is that the outer most points have been contracted inwards due to the noise reduction. This can also be seen in the time history at around 4500 where the peak in the filtered data has been reduced. However this is not surprising since these points are at the edge of the attractor and thus there will be less points in their neighbourhood to define the flow accurately.

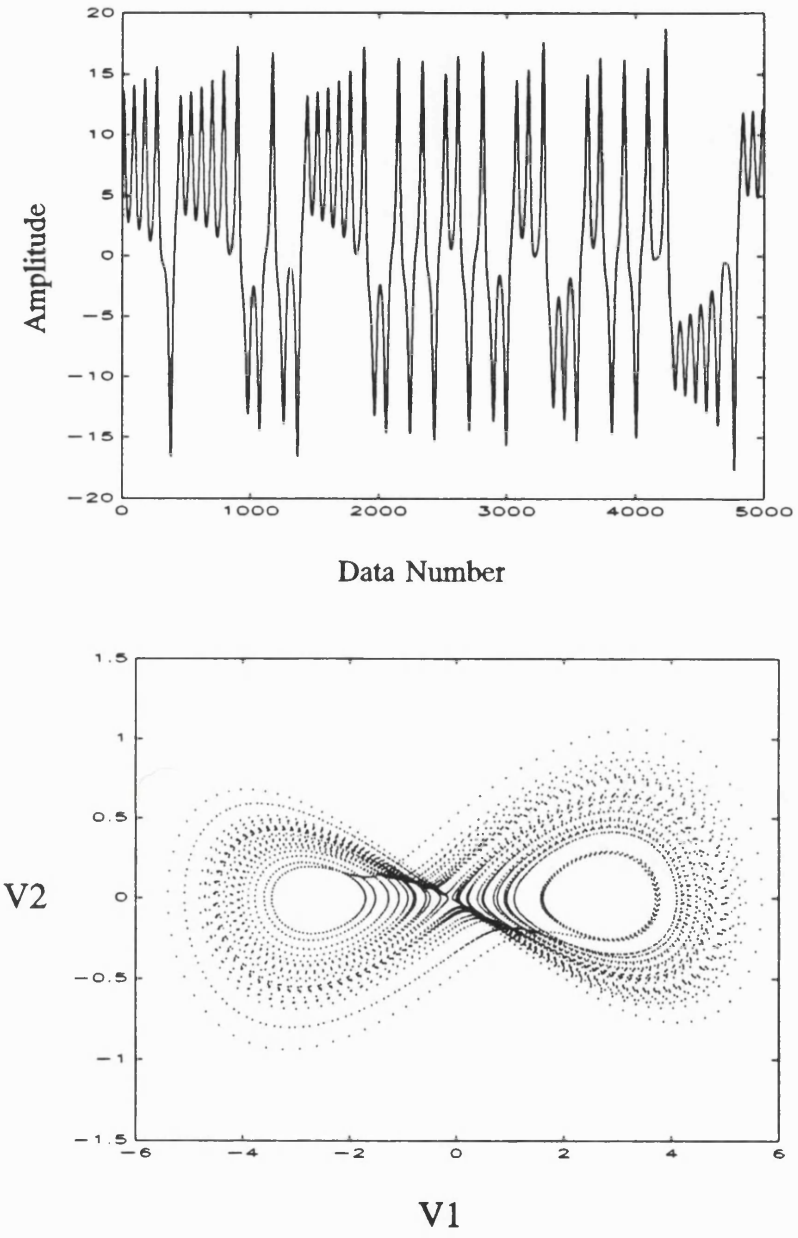


Figure 6.1.1: The top picture shows a 5000 point time series taken from the x coordinate of the Lorenz equations with parameter values: $r = 28$, $b = 8/3$ and $\sigma = 10$. The bottom picture shows the projection of the reconstructed data using the first two singular directions.

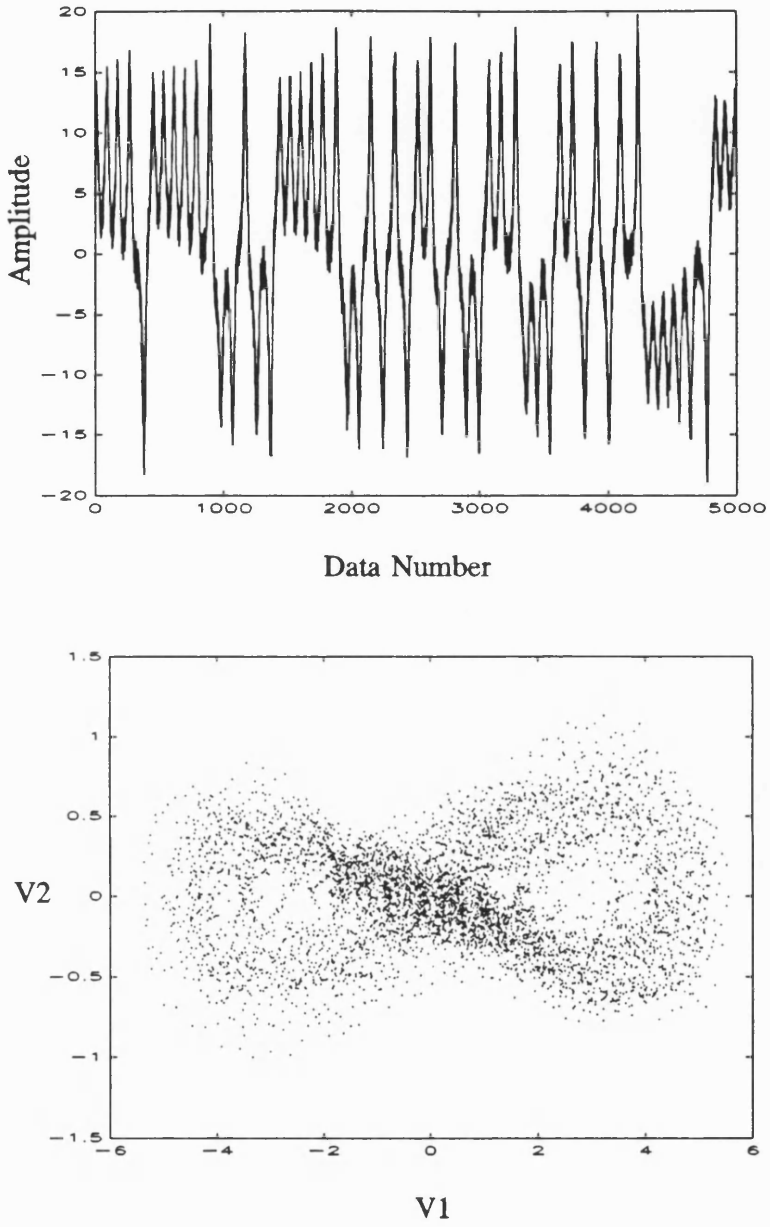


Figure 6.1.2: The top picture shows the data in figure 6.1.1 with 10% noise added. The noise is I.I.D. with uniform distribution. The bottom picture shows the projection of the reconstructed data using the first two singular directions.

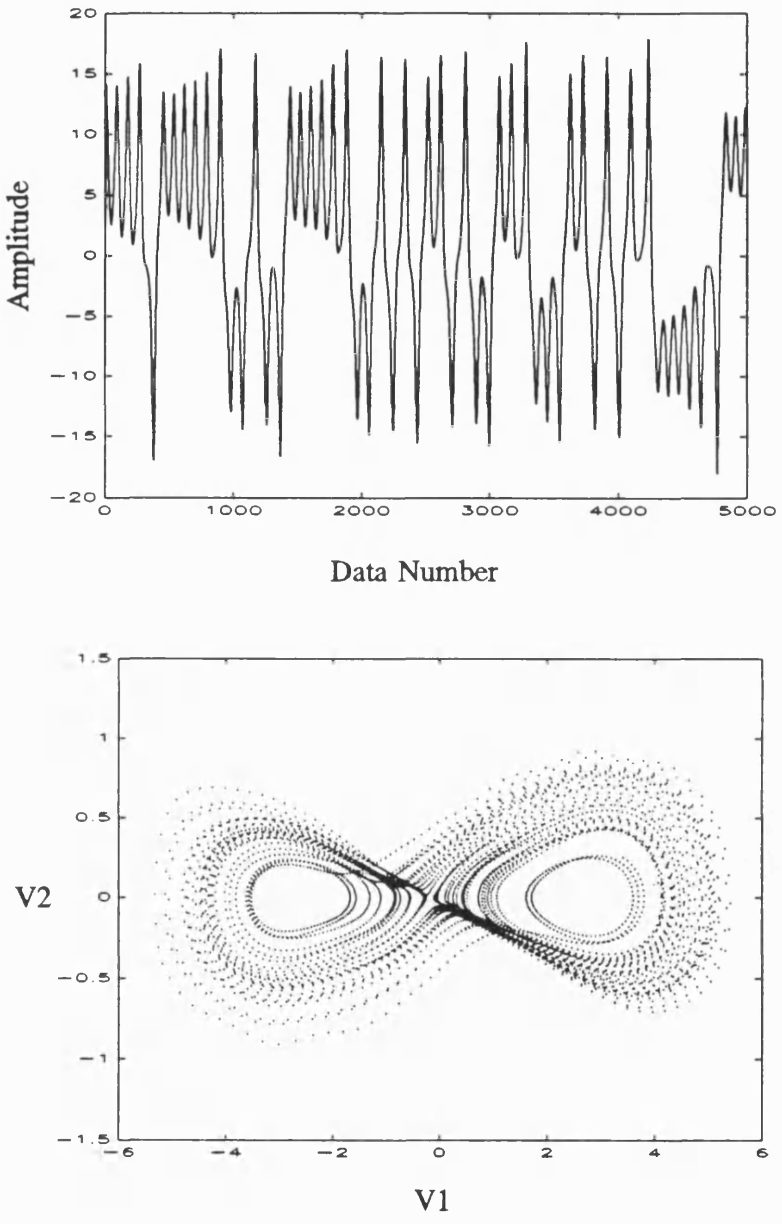


Figure 6.1.3: The top picture shows Lorenz data that was corrupted with 10% noise and then filtered 10 times using the extended Levenberg-Marquardt algorithm. The bottom picture shows the same data reconstructed using the first two singular directions of the trajectory matrix.

We can rate the performance of this noise reduction by looking at the measurement error as well (equation 4.2.1, section 4.2). The original measurement error between the noisy time series and the deterministic orbit was 5.48×10^{3}. After applying noise reduction the error was reduced to 3.25×10^{2}. Thus we have achieved an improvement of just over an order of magnitude with respect to the original data.

The second test we performed was to add 50% noise (in the same sense as above) to the original deterministic orbit. In this case the a great deal of the detail of the data has been lost (figure 6.1.4). Although there is still evidence of the orbit switching between two states, the mechanism by which this happens has been lost. Furthermore when the data is plotted in the reconstructed space none of the structure is evident to the eye.

The filtered data for this test is shown in figure 6.1.5. One immediately obvious observation that can be made is that all the high frequency noise has been removed (to smooth the time series this much with a conventional low pass filter would have distorted the orbit much more) and the trajectory looks smooth again. The state space shows that the attractor has been severely distorted in comparison with the bottom picture in figure 6.1.1. However the basic structure of the attractor has still been captured. The flow direction is still discernable for most points in the embedded space and the two foci of the attractor are clearly recognizable. The dynamic error was 60.4 before noise reduction and this was reduced to 1.64×10^{-3} which is roughly the same improvement that was achieved with 10% noise. The measurement noise before filtering was 1.37×10^{5} . After filtering this was reduced to 7.23×10^{3} which is, again, an improvement of about an order of magnitude. Thus, even when the noise level is high, the algorithm does an impressive job of filtering the data towards a deterministic orbit.

6.1.2 Data from a Periodic Orbit

So far we have described the noise reduction method as one for application to chaotic data sets. However the algorithm is only based on the idea of low dimensional determinacy and it is equally valid to apply it to reduce the noise in non-chaotic data sets. There are several parameter values at which periodic orbits exist in the Lorenz equations

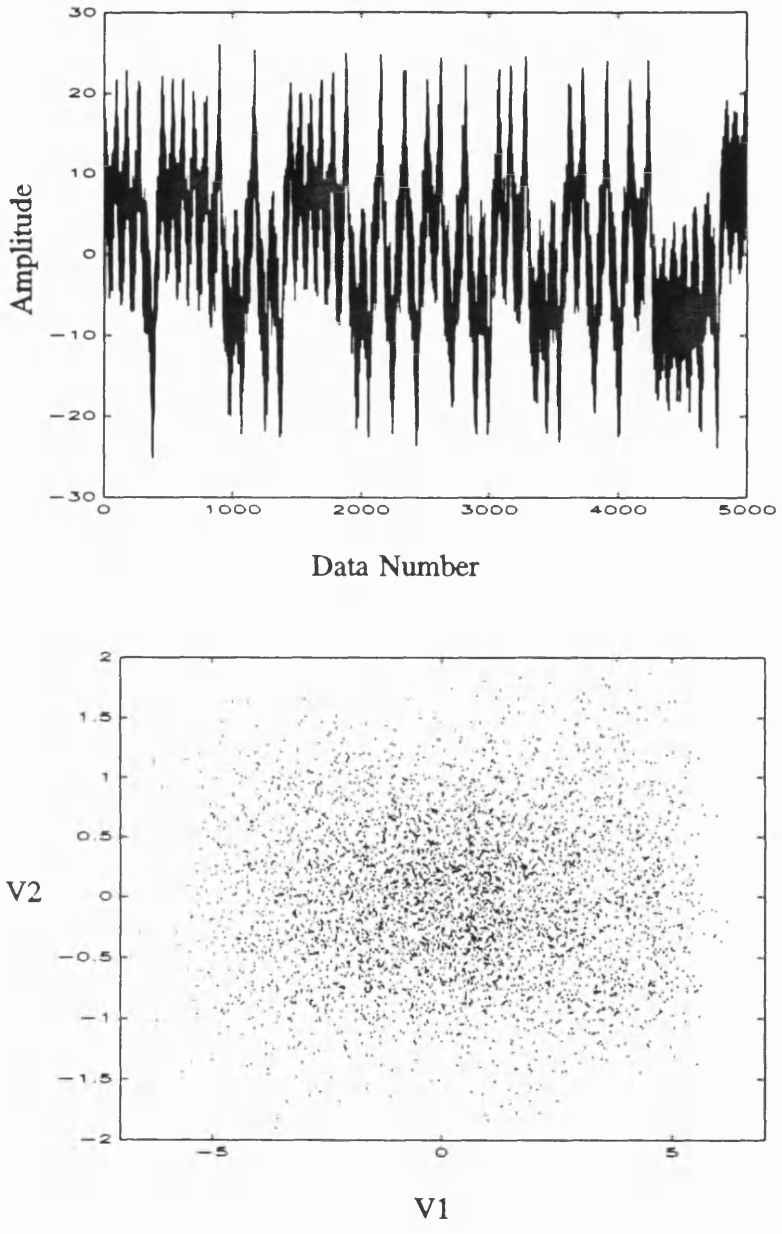


Figure 6.1.4: The top picture shows the data in figure 6.1.1 with 50% noise added. The noise is I.I.D. with uniform distribution. The bottom picture shows the projection of the reconstructed data using the first two singular directions.

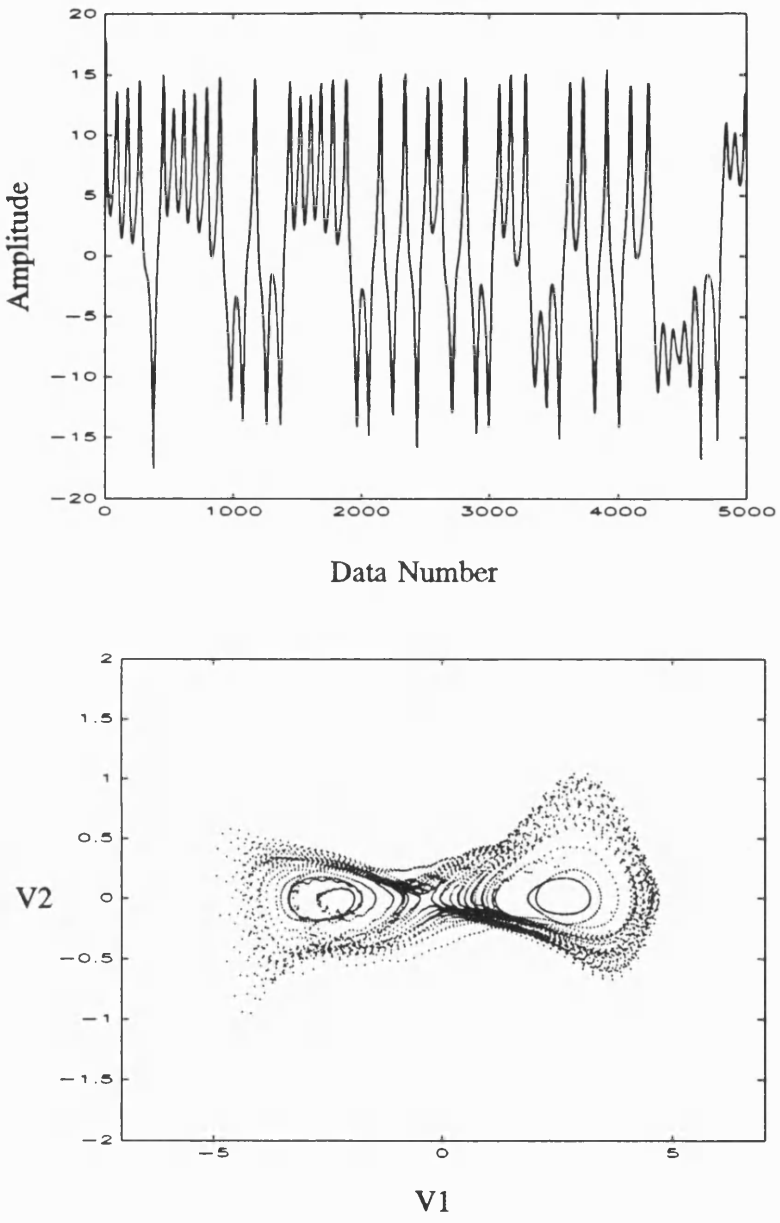


Figure 6.1.5: The top picture shows Lorenz data that was corrupted with 50% noise and then filtered 10 times using the extended Levenberg-Marquardt algorithm. The bottom picture shows the same data reconstructed using the first two singular directions of the trajectory matrix.

(see Sparrow [1982]). Here we use the parameter values: r = 100.5, b = 8/3 and σ = 10. The periodic orbit is shown in figure 6.1.6. along with the same orbit after 10% noise had been added. We then applied 10 iterates of the noise reduction algorithm as above. This orbit is shown in the top picture in figure 6.1.7. Before filtering the time series the dynamic error for the approximated map was: 4.34 whereas after noise reduction this was reduced down to 1.42×10^{-4}. Thus, in this case the reduction in dynamic error about the same for the chaotic attractor with 10% noise and for the limit cycle with 10% noise.

The improvement in measurement error was much greater than for the chaotic attractor. Before the noise reduction it was 1.42×10^{4} and afterwards it was reduced to 6.44×10^{2}. This is an increase in signal-to-noise ratio of approximately 50 times. This improvement relative to the chaotic example is probably due to the fact that the density of points defining the mapping in any given region for the limit cycle is likely to be higher.

6.1.3 A Comparison with Linear Filtering

Although the performance of the noise reduction scheme in the last two sections appears to be excellent this is a little misleading. This is because the signal we are cleaning is predominantly low frequency, whereas the additive noise is chosen to be broad band. To be fair, we should therefore compare the performance of our noise reduction method with that of a traditional linear filter. Here we compare it to a 15th order low-pass Butterworth filter with the cut-off frequency set at the point where the noise floor becomes visible in the power spectrum of the noisy data (Other comparisons between nonlinear noise reduction and linear filters can be found in Kostelich and Yorke [1990] and Grassberger et al [1992]). A butterworth low-pass filter is commonly used in linear signal processing and has the following frequency response function, $H(\omega)$:

$$|H(\omega)| = \frac{1}{\sqrt{1 + \left[\omega / \omega_c\right]^{2N}}} \qquad \textbf{(6.1.2)}$$

where ω_c is the cut-off frequency and N is the order of the filter. For the periodic orbit in the last section the cut-off frequency was chosen to be 0.08, where a frequency of 1.0

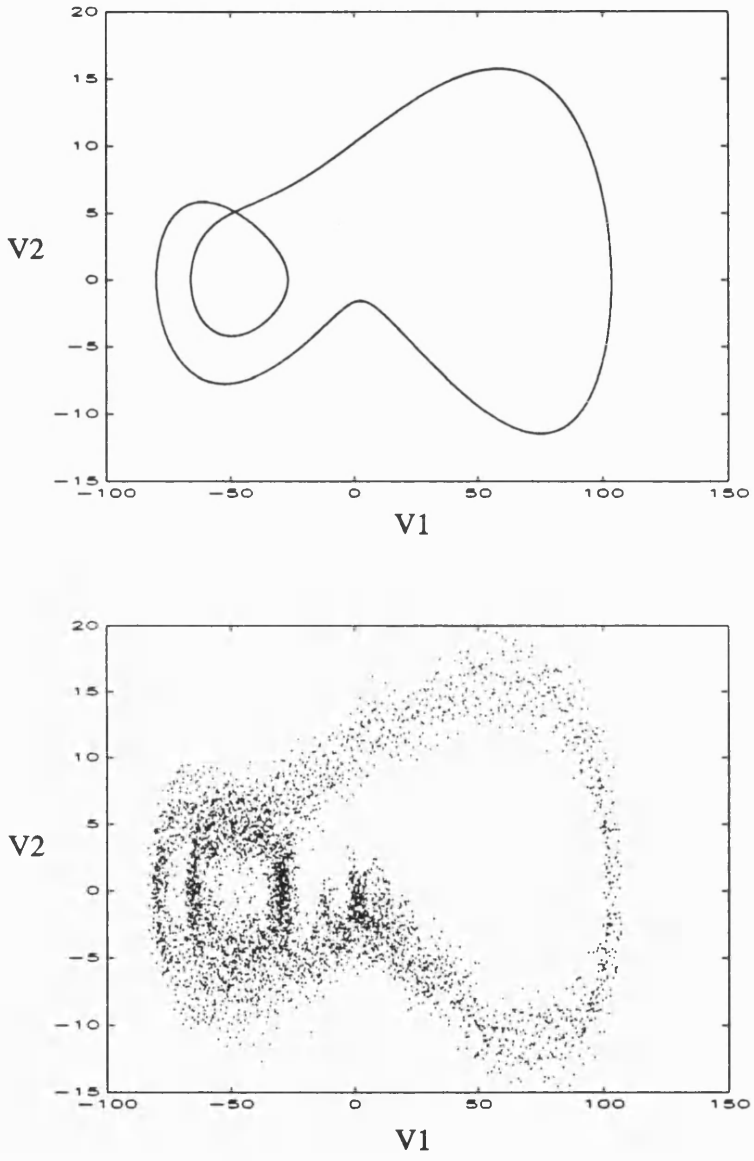


Figure 6.1.6: The top pictire shows the plot of a limit cycle from the Lorenz equations using the first two singular directions.. The bottom picture shows the same data with 10% additive noise.

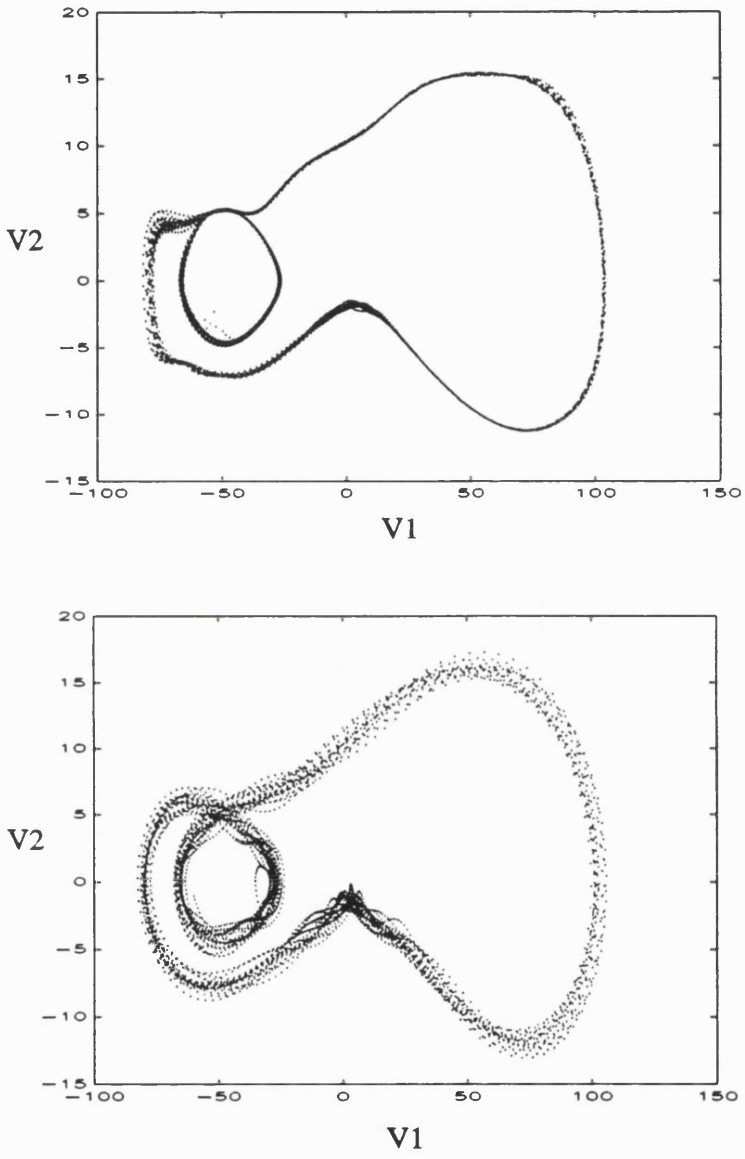


Figure 6.1.7: The top picture shows a singular systems plot of the noisy limit cycle filtered with the Levenberg-Marquardt algorithm. The bottom picture shows the same signal filtered with a 15th order Butterworth low pass filter.

corresponds to half the sampling rate. The state space plot for this filtered orbit is given in the bottom picture of figure 6.1.7 and a comparison of the power spectra for the noisy and clean orbits is given in figure 6.1.8. From this it is clear that the low-pass filter has successfully removed the high frequency noise floor. However the state space plot (figure 6.1.7) allows us to see the difference between the two filtered signals more clearly. The low-pass filter has merely smoothed the orbit. This has not produced a clear periodic orbit. In contrast our noise reduction algorithm concentrates on making the signal deterministic. Hence, although our filtered orbit is more deterministic than the linearly filtered one the power spectrum contains more energy at the higher harmonics (see figure 6.1.8).

Although our noise reduction method has produced a more periodic orbit this does not mean that it is necessarily closer to the original clean orbit. These comparisons are made in the table below. We also compare the measurement error for the two chaotic signals considered in section 6.1.1. For the chaotic signal with 10% noise the cut-off frequency was set at 0.08 and for the chaotic signal with 50% noise the cut-off was 0.06

We should mention that a Butterworth filter is an IIR linear filter an therefore we do not expect it to preserve the dynamics. This also has the effect of introducing transients at the ends of the filtered time series that degrade the signal. To overcome this we compare the all the signals both with and without the end effects (the end effects were removed by simply excluding the first and last 50 points from the error calculations). The order of the Butterworth filter was chosen as a compromise between producing a filter with a sharp frequency cut-off and minimising the end effects.

Below we list the measurement error (with and without end effects) for the three cases examined above as well as the performance of the relevant linear filter in each case.
It is clear from this table that the end effects play an important part in the performance of the Butterworth filter. However, even taking the end effects into account, we can see that our nonlinear noise reduction procedure performs better than the linear filtering in each case.

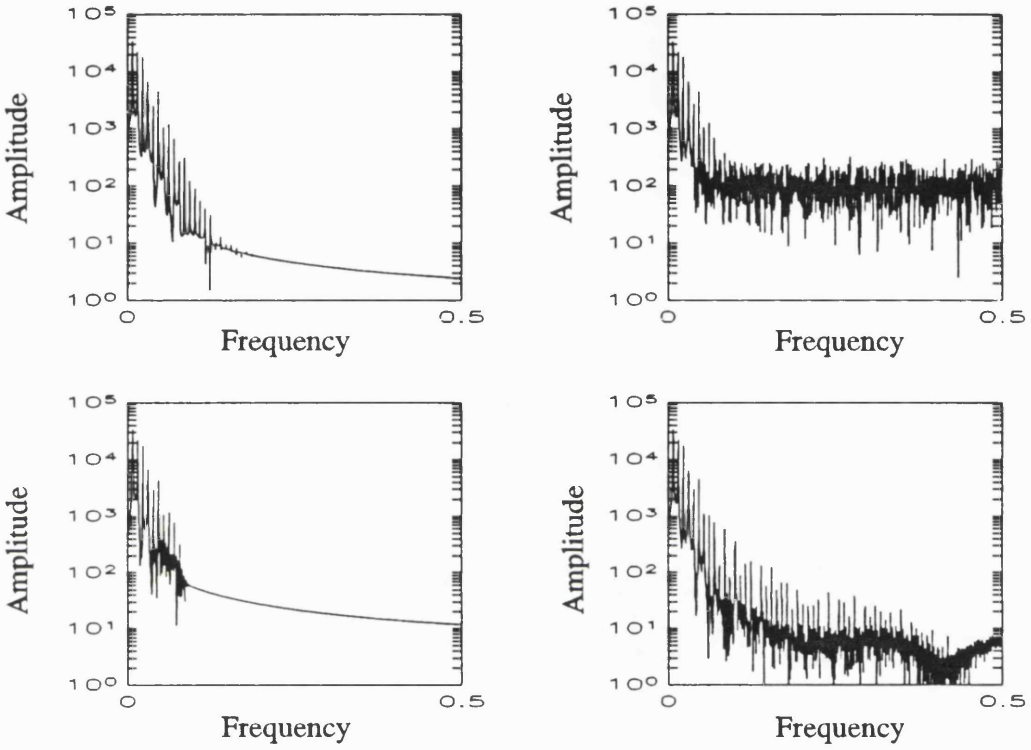


Figure 6.1.8: This plot shows the frequency spectra for, clockwise from the top left: the clean limit cycle, the noisy limit cycle, the noisy limit cycle filtered using a nonlinear noise reduction method and the noisy limit cycle filtered with a linear 15th order low-pass Butterworth filter.

Signal (initial Noise level)	Measurement Error	
	With End Effects	End Effects Removed
Raw Chaotic Signal (10%)	5.48×10^{3}	5.36×10^{3}
Chaotic Signal (10%) after application of noise reduction	3.25×10^{2}	3.06×10^{2}
Chaotic Signal (10%) after filtering with a 15th order Butterworth filter	1.14×10^{3}	4.44×10^{2}
Raw Chaotic Signal (50%)	1.37×10^{5}	1.34×10^{5}
Chaotic Signal (50%) after application of noise reduction	7.23×10^{3}	6.67×10^{3}
Chaotic Signal (50%) after filtering with a 15th order Butterworth filter	3.17×10^{4}	2.71×10^{4}
Raw Limit Cycle (10%)	1.42×10^{4}	1.40×10^{4}
Limit Cycle (10%) after application of noise reduction	6.44×10^{2}	6.13×10^{2}
Limit Cycle (10%) after application of 15th order Butterworth filter	7.14×10^{3}	1.11×10^{3}

6.2 Signal Separation

In the last section we investigated how well our noise reduction algorithm performed on data taken from the Lorenz equations. However an alternative application for the algorithm is to remove an unwanted dynamical signal that swamps some desired ,but unpredictable signal. This reverses the roles of the dynamics and the 'noise' since the noise is now the desired signal and we wish to throw away the dynamics.

Before we consider the performance of the algorithm for this application we must reinterpret the assumptions that need to be made when considering the applicability of this method in light of the role reversal of the signals. Obviously, as we have just stated, the unwanted signal is assumed to come from a low dimensional dynamical system. This idea was originally investigated by Taylor [1992] who was trying to extract a speech signal

that was swamped by air-conditioning noise. Similarly the desired signal must be unpredictable and possess certain statistical properties, such as stationarity. However the most counter-intuitive assumption is that the desired signal should be small in relationship to the dynamical signal. This is unusual since it would seem to be more natural if the algorithm performed better when the desired signal was larger.

Finally it is necessary to re-evaluate the performance criterion by which the algorithm is judged. Here we calculate the signal-to-noise ratio of the stochastic signal before and after noise reduction. The signal-to-noise of the predicted signal will simply be a function of the signal-to-noise of the noisy data and the signal-to-noise of the estimate for the deterministic signal. Thus we can re-evaluate the three tests done above in terms of how well the noise could be estimated. The table below lists the signal-to-noise ratio for the estimates in each case. In each case the signal-to-noise was increased to around 4.0 and it is interesting that this seems not to be particularly sensitive on the initial noise level.

Data Set	Original SNR	Final SNR
Chaotic Attractor with 10% noise	0.131	4.10
Chaotic Attractor with 50% noise	0.658	4.35
Periodic Attractor with 10% noise	0.105	4.70

6.3 Improved Deterministic Modelling: Experimental Laser Data

One problem that can be encountered from modelling the dynamics of a time series is that the model may not possess the same asymptotic properties of the data. Although we can never really expect to capture the original attractor's properties exactly (especially if it is non-hyperbolic) we would like to at least obtain an attractor that looked similar to the data set (i.e. such that the probability distributions are in some sense similar). For example we would not want to chose a model for a chaotic data set to have a limit cycle

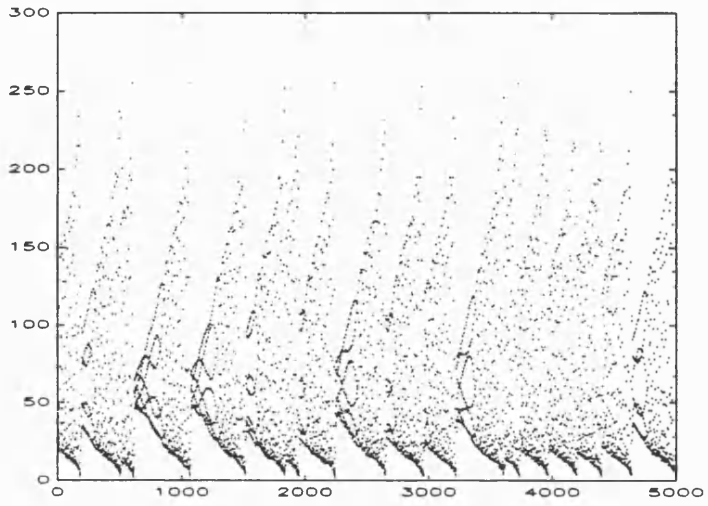
as its asymptotic attractor.

At the end of chapter 4 we discussed the possibility of using noise reduction to improve the model of the dynamics. The basic argument is as follows. Statistically, if we have modelled the dynamics with a one step error term, it is necessary to retain this stochastic component of the system when investigating the system further. Indeed this term is essential when fitting linear state space models to data since, in this case, without the noise component the dynamics would be uninteresting. To throw away the noise term requires an argument that the dynamics are unaffected by the noise. Although we have this in Bowen's shadowing lemma for hyperbolic systems even here the levels of allowable noise may be unrealistically small. Hence we argue that reducing the size of the stochastic term improves the chances that the stochastic term may be neglected. We have already seen in chapter 4 that such noise reduction is made easier by including the flexibility of the function approximation into the noise reduction process. Thus we have an algorithm by which we can obtain a function approximation that reduces the required stochastic component.

We can test this argument by comparing the deterministic attractors of the approximated dynamics, before and after noise reduction, with the original data to see which system models the data's asymptotic behaviour best. Although we could calculate various invariants of the data, here we will merely discuss the overall qualitative shape of the attractor.

We applied this idea to some experimental data taken from a laser. The data was originally collected by N.B. Abraham and C.O. Weiss from a far-infrared laser running in a chaotic state. The time series recorded was some measurement of intensity of the laser. For further details on the data and the experimental set up see Hübner et al [1989a, 1989b and 1989c]. The initial 1000 data points were later used to test prediction models in the Santa Fe Institute time series prediction and analysis competition and we obtained the complete 10000 point data set from the Santa Fe *ftp* site.

Here we took the first 5000 points from this time series as our data set. The time history is shown in the top picture in figure 6.2.1. However the observable measured was



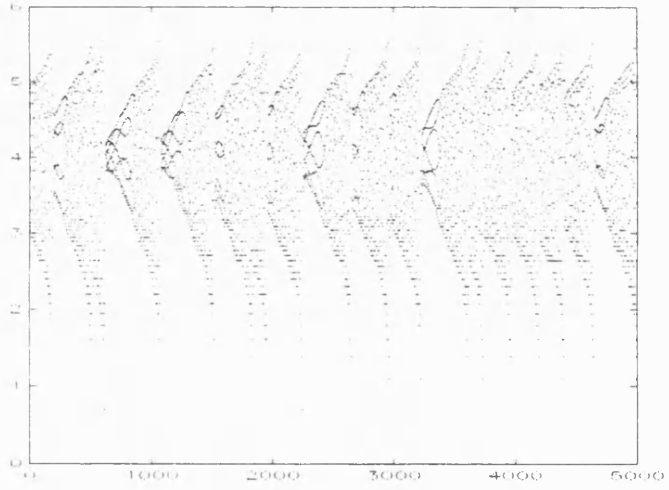


Figure 6.2.1: This plot shows the time history of a power measurement from a laser (top) and the logorithm of the same data (bottom).

intensity which, Smith [1993] noted was in some sense degenerate since it is the square of some measurement and may therefore project two different parts of the attractor on top of each other. Here we assume that the dynamics are symmetric about the origin so that this does not matter. Even so this still leaves the data constrained to be greater than or equal to zero. A standard function approximation on this data would not take this into account since a small error resulting in a prediction less than zero is treated in the same way as one that produces a positive predicted value. Thus we are really confronted with a constrained optimisation problem. A crude way to rectify this problem is merely to use the logarithm of the data: i.e. to map $\mathbf{R}^+ \rightarrow \mathbf{R}$. This is the time series that we used and it is shown in the bottom picture in figure 6.2.1.

To approximate the dynamics we then embedded the logged data (rescaled to [-0.5,0.5]) in a 3 dimensional state space using singular systems analysis with a delay window length of 10. Then we estimated the mapping function with 100 Gaussian radial basis functions with a rescaling parameter set at 1.0, choosing the centres from the data set as usual. The weights were initially estimated using an ordinary least squared estimation, as described in section 2.5.2.

We then applied the Levenberg-Marquardt algorithm proposed in section 4.2.2 with the Levenberg-Marquardt parameter set at 1.0. This was iterated 10 times and the dynamic error was reduced from 3.10 down to 8.08×10^{-3}. The resulting estimates for the weights in the model for the mapping function provide us with a second model for the dynamics. such that a nearby time series is modelled with less error than the ordinary least squared estimate models the original time series.

The original embedded data is shown in the top picture of figure 6.2.2 plotting the 2nd singular direction against the 3rd (these were chosen since they provided a better view of the oscillations than using the 1st and the 2nd singular directions). The repeated spiralling out from the centre as the oscillations grow is clearly visible although the re-injection mechanism cannot be discerned from this projection.

We can compare this embedded data to a 5000 point trajectory produced by iterating the approximation model embedded in the same way. This is shown in the bottom picture in

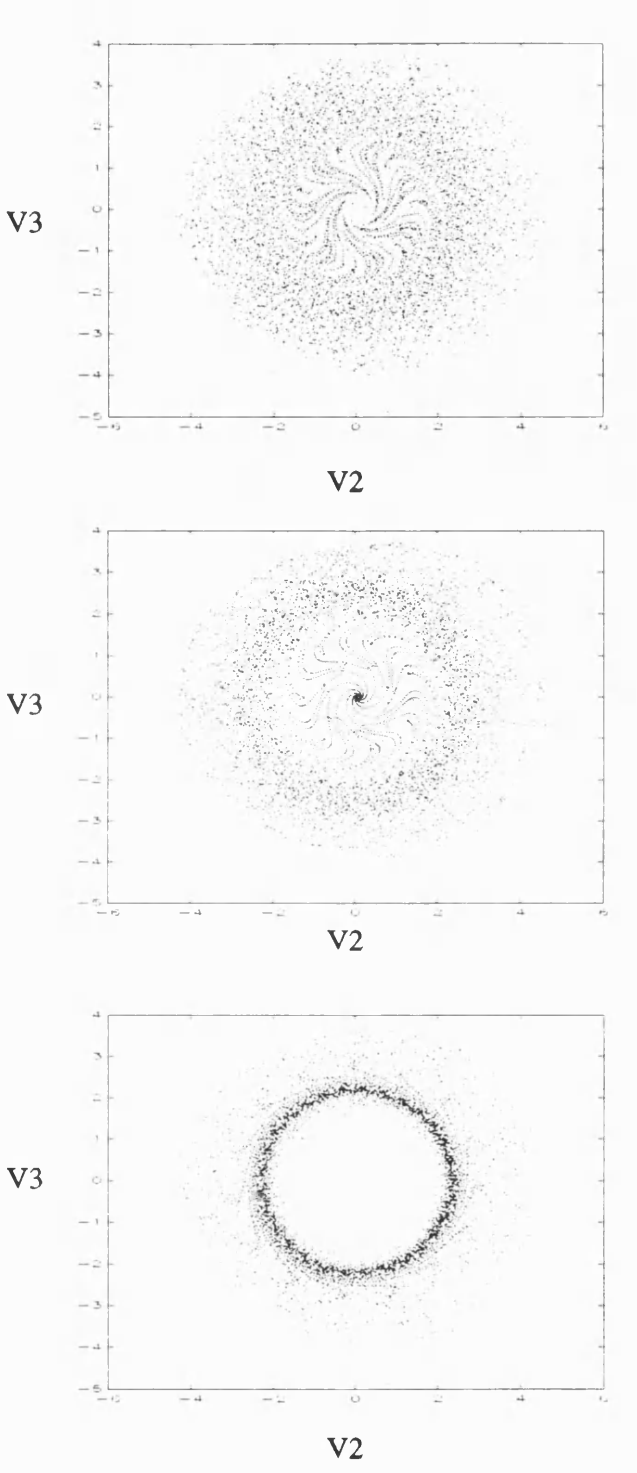
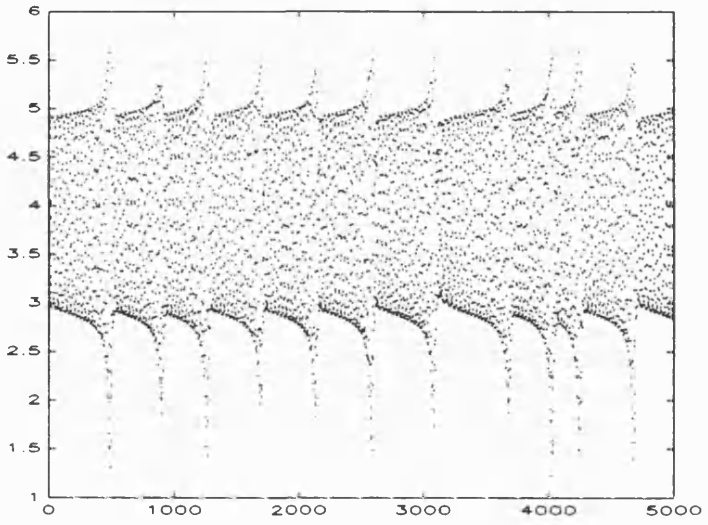


Figure 6.2.2: This plot compares the attractor for the noisy data (top) to the attractor for the approximated system using a least squared estimate (bottom) and the attractor for the approximated system after noise reduction (middle).

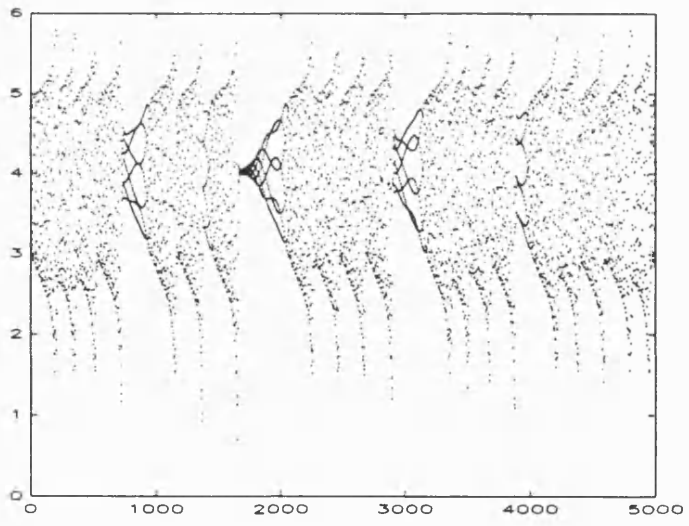
figure 6.2.2. We would hope that this data set would duplicate the repeated growth and subsequent sudden fall in the amplitude of the oscillations. However the plot in figure 6.2.2 shows that the amplitudes of the oscillations remain high. Although there is still some variation in amplitude and the attractor of the approximated system appears to be chaotic it is clear that the mechanism by which the trajectory was injected into the middle of the attractor has been lost.

In comparison, if we take a 5000 point trajectory iterated from the model resulting from the noise reduction process we obtain a far better deterministic model for the original data. The middle plot in figure 6.2.2 shows this trajectory plotted in the same way as the other two. The re-injection process has obviously been modelled far more successfully since this system does appear to duplicate the repeated spiralling out from the centre. Finally we can see the difference between the two iterated systems in figure 6.2.3 where the top picture shows the time history from the initial estimate for the dynamics and the bottom picture shows the time history from the adjusted system after noise reduction. In the top picture the slow growth in the amplitude of the followed by rapid decline is present but the amplitude does not vary nearly as much as the original data. In contrast, both of these features are excellently captured in the bottom picture.

We are not claiming here that an ordinary least squared estimate could not have achieved such a good a model. Indeed, Sauer [1993] has produced a model that successfully models the behaviour of the data, based only on the first 1000 data points. However we are arguing that for a given function form the asymptotic behaviour of the model is likely to be improved after noise reduction. This of course assumes that the function approximation is in some way incorporated into the noise reduction process.



Iterate Number



Iterate Number

Figure 6.2.3: This plot shows time histories from the approximated dynamical system both before (top) and after (bottom) noise reduction is applied.

7. Conclusions

Over the last decade a variety of new techniques for the treatment of chaotic time series has been developed. Initially these concentrated on the characterisation of chaotic time series using invariants such as fractal dimensions or Lyapunov exponents. However, attention has recently focused on the possibility of predicting the future short term behaviour of such time series using the type of methods discussed in chapter 2. It soon became obvious that these methods could be used to develop a set of novel signal processing tools based on nonlinear state space theory. This in turn resulted in the development of a proliferation of noise reduction algorithms to clean chaotic time series that had been corrupted by low amplitude noise. Although this is by no means the only application of nonlinear dynamics to signal processing and many other directions are being pursued by a number of researchers it seems to have been the direction that has gained most attention. Thus this is the problem we have concentrated on here.

Initially we described the noise reduction methods that are based loosely on the shadowing lemma. This type of algorithm falls into two basic groups: those that achieve the shadowing explicitly and those that achieve it implicitly. In section 3.2.3 we explained why the extra work required to maximise the shadowing property explicitly is not worth while. This is because the shadowing orbit is 'almost unique' and the only optimisation that occurs in explicit shadowing methods effects the ends of the time series. Furthermore these end effects reduce at an exponential rate as they propagate down along the data and very quickly become insignificant.

The shadowing problem can then interpreted as a rank deficient zero-finding problem. In this context Hammel's implicit noise reduction algorithm applies a set of reasonable constraints to the problem to make it full rank and in section 3.2 we discussed how these constraints could be calculated in practice. However this does not address the possibility of there being homoclinic tangencies present in the dynamics. In section 3.2.3 we showed that these are extremely important to the performance of a noise reduction algorithm since they make the root-finding problem ill-conditioned (even with the extra constraints added). One way to circumvent this problem is to break the time series up into smaller lengths of data such that the root-finding problem is well conditioned for each length

individually. However this is not ideal and we were keen to develop a more systematic method of dealing with the presence of tangencies.

The approach we have taken here has been to produce a new set of noise reduction algorithms based on minimising the dynamic error. This is a natural choice since it is one of the criteria used to judge the performance of a noise reduction algorithm and it is the only criterion that is readily available without *a priori* knowledge of the true solution. This can also be shown to be the minimisation equivalent of the root-finding problem solved by Hammel. However the one difference is that this approach now allows us to introduce the idea of weak shadowing: i.e. being able to move towards the solution without directly guessing where it is. This enabled us to compare in a precise way many of the different types of noise reduction algorithm that have recently been proposed.

One very simple noise reduction method to come out of this idea is the solution by gradient descent which we introduced in section 3.4.2. This requires minimal computational effort and can be made to be as stable as necessary, while guaranteeing to reduce noise. It also turns out that this method relates strongly to various *ad hoc* approaches that have been proposed, based on projecting the embedded data towards some approximation of the deterministic manifold. Some of these can then be reformulated to show that they are equivalent to minimising some error function by a gradient descent method. Furthermore these methods are a good way to tackle the presence of tangencies and we show that solving the minimisation problem using a gradient descent algorithm effectively 'avoids' the tangencies by simply not stepping in the relevant directions.

However the one drawback with the gradient descent algorithm is its speed. The gradient descent algorithm can be shown to converge onto the minimum at a linear rate, whereas a Newton based algorithm (for example Hammel's method suitably transformed into the minimisation context) will converge at a quadratic rate. Thus neither Newton based methods nor gradient descent methods are ideal since Newton methods become unstable with tangencies and gradient descent methods are slow. To produce an acceptable compromise we applied the Levenberg-Marquardt algorithm to the minimisation problem. This provides a systematic method by which it is possible to interpolate between Newton

and gradient descent based methods and as we have shown we can use the Levenberg-Marquardt parameter to inflict an upper bound on the condition of the problem. The incentive for using the Levenberg-Marquardt algorithm is thus to find an algorithm with approximately quadratic convergence that remains stable. In chapter 5 we numerically demonstrated that we were able to achieve this in practice.

In chapter 4 we went on to extend these ideas to include the problem of modelling the dynamics within the noise reduction algorithm. This makes sense since both procedures aim to minimise the dynamic error (which is identical to the approximation error in the function approximation), but with respect to different parameters. Looked at in these terms is seems inefficient to repeatedly apply an iteration of the noise reduction algorithm followed by refitting the dynamics. It makes more sense to solve them together, making use of the fact that the problems are linked. We explored this idea and found that it was possible to produce an extended Levenberg-Marquardt algorithm that solved the two problems simultaneously. When this approach was compared to the two step approach we found it to be both more stable as well as producing a filtered orbit closer to the original data.

Having developed a workable algorithm, we made a number of numerical investigations to demonstrate the performance of our noise reduction algorithm. In chapter 6 we demonstrated that our method was easily adaptable to more general embedding spaces and quite high levels of noise. We then compared our method to a traditional linear filter and in each case our method proved to be superior. These results were also interpreted in terms of removing the dynamics from the noise and again the algorithm was shown to perform well. Thus we can view the algorithm as separating the input signal into two parts. It is then up to the user to decide which signal to keep and which to throw away.

Finally, as well as discussing the obvious role reversal possible for the dynamics and noise, we introduce a novel use of these noise reduction methods to achieve an improved deterministic model as a by-product of the filtering process. The main argument being that traditional methods for modelling the dynamics optimise the one-step prediction whereas the model that results from our noise reduction scheme is designed to produce good shadowing properties. In the last section of chapter 6 we demonstrate this idea on

some experimental laser data. Here the initial model for the dynamics did not produce an attractor that was similar to the embedded data whereas the model for the dynamics resulting from the noise reduction appeared to accurately capture the characteristics of the original data set.

References:

Alekseev, V. M. and Yakobson, M.V., 1981 Symbolic dynamics and hyperbolic dynamical systems. *Physics Reports*, **75**, No.5:287-325.

Aleksic, Z. 1990 Estimating the embedding dimension. *Technical Report, Visual Science Centre, Australian National University, Canberra, ACT2601.*

Badii, R., Broggi, G., Derighetti,B., Ravani, M., Ciliberto, S. Politi, A. and Rubio, M.A. 1989 *Phys. Rev. Lett.*, **60**:979.

Bentley, J. L. and Freidman, B. H. 1979 Data structures for range searching. *Computing Surveys*, Vol **11**, No. 4:397-409.

Bowen, R. 1970 Markov partitions for Axiom A diffeomorphisms. *Amer. J. Math.*, **92**:725-747.

Bowen, R. 1978 *On Axiom A Diffeomorphisms*. CBMS Regional Conference Series in Mathematics, Vol **35**. A.M.S. Publications.

Bröcker, TH. and Janich, K. 1982 *Introduction to differential topology*. Cambridge University Press.

Broomhead, D.S. and King, G.P. 1987, Extracting qualitative dynamics from experimental data. *Physica D*, **20**:217-236.

Broomhead, D.S., Jones, R. and King, G.P. 1987 Topological dimension and local coordinates from time series data. *Journal of Physics A*, **20**:563-569.

Broomhead, D.S. and Lowe, D. 1988 Multivariate functional interpolation and adaptive networks. *Complex Systems*, **2**:321-355.

Broomhead, D.S., Huke, J.P. and Muldoon, M.R. 1992 Linear filters and nonlinear systems. *Journal Stat. Phys.*, **65**.

Brown, R., Bryant, P. and Abarbanel, H.D.I. Computing the Lyapunov spectrum of a dynamical system from an observed time series. *Phys. Rev. A*, **43(6)**:2787-2806.

Casdagli, M., 1989 Nonlinear prediction of chaotic time series. *Physica D*, **35**:335-356.

Casdagli, M., Eubank, S., Farmer, J.D. and Gibson, J. 1991 State space reconstruction in the presence of noise. *Physica D*,**51**.

Cawley, R and Hsu, G.-H. 1992 SNR performance of a noise reduction algorithm applied to coarsely sampled chaotic data. *Phys. Lett. A*, **166**:18-196.

Cenys, A. and Pyragras, K. 1988 Estimation of the number of degrees of freedom from chaotic time series. *Phys. Lett. A*, 129:227.

Constantin, P., Foias, C., Nicolaenko, B. and Temam, R. 1989 *Integral Manifolds and Inertial Manifolds for Dissipative Partial Differential Equations*. Applied Mathematical Sciences **70**, Springer-Verlag.

Cremers, J. and Hubler, A. 1987 Construction of differential equations from experimental data. *Z.Naturforsch*, **42a**:797-802.

Davies, M. 1992 An Iterated function approximation in shadowing time series. *Phys. Lett. A* **169**:251-258.

Davies, M. 1992 Noise Reduction by Gradient Descent. *International Journal of Bifurcation and Chaos*, Vol.3 No.1:113-118.

Davies, M. 1992 An Order N Noise Reduction Algorithm with Quadratic Convergence. *Submitted to Physica D.*

Eckmann, J-P. and Ruelle, D. 1985 Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, **57(3)**:617-656.

Farmer, J. D., Ott, E. and Yorke, J. A. 1983 The dimension of chaotic attractors. *Physica D*, **7**:153-180.

Farmer, J.D. and Sidorowich, J.J. 1987 Predicting Chaotic time series. *Phys. Rev. Lett.*, **59(8)**:845-848.

Farmer, J.D. and Sidorowich, J.J. 1988 Exploiting Chaos to Predict the Future and Reduce Noise. in *Evolution, Learning and Cognition, ed. Y.C. Lee* World Scientific, Singapore.

Farmer, J.D. and Sidorowich, J.J. 1991 Optimal shadowing and noise reduction. *Physica D*, **47**:373-392.

Fletcher, R. 1980 Practical Methods of Optimisation, Vol 1. Unconstrained Optimisation. *John Wiley and Sons.*

Fletcher, R. 1981 Practical Methods of Optimisation, Vol 2. Constrained Optimisation. *John Wiley and Sons.*

Fraser, A. M. and Swinney, H. L. 1986 Independent coordinates for strange attractors from mutual information. *Phys. Rev. A* **33**:1134-1140.

Fraser, A.M. 1989 Reconstructing attractors from scalar time series: a comparison of singular system and redundancy criteria. *Physica D*, **34**:391-404.

Gibson, J. F., Farmer, J. D., Casdagli, M. and Eubank, S 1992 An analytic approach to practical state space reconstruction. *Physica D*, **57**:1-30.

Golub, G.H. and Van Loan, C.F. 1983 *Matrix Computations* (Baltimore: John Hopkins University Press).

Grassberger, P. and Procaccia, I. 1984 Dimensions and entropies of strange attractors from a fluctuating dynamics approach. *Physica D*, **13**:34.

Grassberger, P., Hegger, R., Kantz, H., Schaffrath, C. and Schreiber, T 1992 On Noise reduction methods for chaotic data. *Preprint.*

Green, M. and Savit, R., 1992 Dependent Variables in broad band continuous time series. *preprint.*

Guckenheimer, J. and Holmes, P. 1983 *Nonlinear oscillations, dynamical systems and bifurcations of vector fields.* Applied Mathematical Sciences **42**, Springer-Verlag.

Hammel, S. M., Yorke, J. A. and Grebogi, C. 1988 Numerical orbits of chaotic processes represent true orbits. *Bull. Amer. Math. Soc.*, **19(2)**:465-469.

Hammel, S.M. 1990 A noise reduction method for chaotic systems. *Phys. Lett. A*,**148**, No 8,9:421-428.

Hirsch, M. W. 1976 *Differential Topology. Graduate texts in mathematics 33*, Springer-Verlag.

Hübner, U., Abraham, N. B. and Weiss, C. O. 1989 Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser. *Phys. Rev. A* **40**:6354.

Hübner, U., Klische, W., Abraham, N. B. and Weiss, C. O. 1989 On problems encountered with dimension calculations. *Measures of Complexity and Chaos* (Ed. by N. B. Abraham et. al.), Plenum Press, New York.

Hübner, U., Klische, W., Abraham, N. B. and Weiss, C. O. 1989 Comparison of Lorenz-like laser behaviour with the Lorenz model. *Coherence and Quantum Optics VI* (Ed. J. Eberly et. al.), Plenum Press, New York.

Hunt, B. R., Sauer, T. and Yorke, J. A. 1992 Prevalence: a translation-invariant "almost every" on infinite dimensional spaces. *to appear in Bulletin of AMS.*

Johnson, R., Palmer, K. and Sell, R. 1987 Ergodic Properties of Linear Dynamical Systems. *SIAM J. Math Anal.*, Vol **18**, No 1:1-33.

Kennel, M., Brown, R. and Abarbanel, H. 1992 Determining embedding for phase-space reconstruction using a geometrical construction. *Phys Rev A*, **45** No 6:3403-3411.

Kostelich, E.J. and Yorke, J.A. 1990 Noise reduction: finding the simplest dynamical system consistent with the data. *Physica D*, **41**:183-196.

Liebert, W., Pawelzik, K. and Schuster, H. G. 1991 Optimal embeddings of Chaotic attractors from Topological Considerations. *Europhys. Lett.*, **14** (**6**):521-526.

Lorenz, E. N. 1969 Atmospheric predictability as revealed by naturally occurring analogues. *J. Atmospheric Sci.* **26**:636-646.

Meyer, T. and Packard, N. H. 1992 Local Forecastng of High-Dimensional Chaotic Dynamics. *in Nonlinear Modelling and Forecasting* ed. Casdagli M. and Eubank S., 249-264.

Michelli, C. A. 1986 Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. *Constructive Approximation*, **2**:11-22.

Noakes, L. 1991 The Takens' Embedding Theorem. *Int. Journ. of Bifurcation and Chaos*, Vol **1**, No 4:867-872.

Nusse, H. and Yorke, J. 1988 Is Every Approximate Trajectory of Some Process Near an Exact Trajectory of a Nearby Process? *Commun. Math. Phys.* **114**:363-379.

Oseledec, V. I. 1968 A multiplicative ergodic theorem: Lyapunov characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc.*, **96**:17-39.

Packard, N. H., Crutchfield, J. P., Farmer, J.D. and Shaw, R.S. 1980, Geometry from a time series. *Phys. Rev. Lett.*, **45**:712-716.

Powell, M. J. D., 1985 Radial basis functions for multivariable interpolation: a review. *IMA conference on 'Algorithms for the Approximation of Functions and Data'*, RMCS Shrivenham.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. 1992 *Numerical Recipes in C.* Cambridge University Press.

Ruelle, D. and Takens, F. 1971, On the nature of turbulence. *Comm. Math. Phys.*, **20**:167-192.

Sauer, T. 1992 A Noise reduction method for signals from nonlinear systems. *Physica D*, **58**:193-202.

Sauer, T., Yorke, J. A. and Casdagli, M. 1991 Embedology. *Journal Stat. Phys.*, **65**:579-616.

Sauer, T. 1993 Time series prediction using delay coordinate embedding. To appear in: *Predicting the future and understanding the past: a comparison of approaches* (ed. A. Weigend and N. Gersenfeld), Addison-Wesley.

Savit, R. and Green, M. 1991 Time series and dependent variables. *Physica D*, **50**:95-116.

Schreiber, T. and Grassberger, P. 1991 A simple noise reduction method for real data. *Phys. Lett A.* **160(5)** 411-418.

Shaw, R. S. 1985 The Dripping Faucet as a model chaotic system. *(Aerial, Santa Cruz, C.A.).*

Shub, M. 1986 *Global Stability of Dynamical Systems*. Springer-Verlag.

Smith, L. 1993 Does a meeting in Santa Fe imply Chaos? To appear in: *Predicting the future and understanding the past: a comparison of approaches* (ed. A. Weigend and N. Gersenfeld), Addison-Wesley.

Sparrow, C. 1982 *The Lorenz equations: bifurcations, chaos and strange attractors* Applied Mathematical Sciences **41** Springer-Verlag.

Sugihara, G. and May, M. 1990 Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, **344**:734-741.

Takens, F., 1981 Detecting strange attractors in fluid turbulence. *Dynamical Systems and Turbulence, Lecture Notes in Math.* **898** (ed. D. Rand and L-S. Young) Berlin, Springer-Verlag.

Taylor, W. 1992 Application of Nonlinear Prediction to Signal Separation. *in Nonlinear Modelling and Forecasting* ed. Casdagli, M. and Eubank, S. 455-467.

Temam, R. 1988 *Infinite Dimensional Dynamical Systems in Mechanics and Physics.* Applied Mathematics Series, Vol. **68**, Springer-Verlag.

Zak, M. 1989. Terminal Attractors in Neural Networks. *Neural Networks* **2**:259-274.