University of Tennessee, Knoxville

Trace: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

8-2003

Detecting Changes in Global Dynamics with Principal Curves and Information Theory

Sandeep Rajput

University of Tennessee - Knoxville

Recommended Citation

Rajput, Sandeep, "Detecting Changes in Global Dynamics with Principal Curves and Information Theory. " PhD diss., University of Tennessee, 2003.
https://trace.tennessee.edu/utk_graddiss/2380

To the Graduate Council:

I am submitting herewith a dissertation written by Sandeep Rajput entitled "Detecting Changes in Global Dynamics with Principal Curves and Information Theory." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Chemical Engineering.

Duane D. Bruns, Major Professor

We have read this dissertation and recommend its acceptance:

C. Stuart Daw, John R. Collier, Charles F. Moore, Frank M. Guess, J. Wesley Hines

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Sandeep Rajput entitled "Detecting Changes in Global Dynamics with Principal Curves and Information Theory." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Chemical Engineering.

Duane D. Bruns
Major Professor

We have read this dissertation
and recommend its acceptance:

C. Stuart Daw

John R. Collier

Charles F. Moore

Frank M. Guess

J. Wesley Hines

Acceptance for the council:

Anne Mayhew
Vice Provost and Dean of
Graduate Studies

(Original signatures are on file with official student records.)

DETECTING CHANGES IN GLOBAL DYNAMICS WITH PRINCIPAL CURVES AND INFORMATION THEORY

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Sandeep Rajput
August 2003

Abstract

Two approaches to characterize global dynamics are developed in this dissertation. In particular, the concern is with nonlinear and chaotic time series obtained from physical systems. The objective is to identify the features that adequately characterize a time series, and can consequently be used for fault diagnosis and process monitoring, and for improved control.

This study has two parts. The first part is concerned with obtaining a skeletal description of the data using Cluster-linked principal curves (CLPC). A CLPC is a non-parametric hypercurve that passes through the center of the data cloud, and is obtained through the iterative Expectation-Maximization (E-M) principle. The data points are then projected on the curve to yield a distribution of arc lengths along it. It is argued that if some conditions are met, the arc length distribution uniquely characterizes the dynamics. This is demonstrated by testing for stationarity and reversibility based on the arc length distributions.

The second part explores the use of mutual information vector to characterize a system. The mutual information vector formed via symbolization is reduced in dimensionality and subjected to K-means clustering algorithm in order to examine stationarity and to compare different processes.

The computations required to implement the techniques for online monitoring and fault diagnosis are reasonable enough to be carried out in real time. For illustration purposes time series measurements from a liquid-filled column with an electrified capillary and a fluidized bed are employed.

Keywords: *Chaos, clustering, fault diagnosis, information theory, monitoring, mutual information, nonlinear dynamics, principal curves, process control, symbolization*

Acknowledgements

I would like to express my deep gratitude to my advisor, Duane D. Bruns, for encouragement, trust, professional support, and invaluable advice on things chaotic and linear. I would also like to thank my dissertation committee for their insights and support. In particular, C. Stuart Daw was edifying in discussions about nonlinear dynamics and chaos; Frank Guess provided many helpful comments about some statistical aspects of this dissertation; J. Wesley Hines pointed out relevant research; and John R. Collier and Charles F. Moore posed leading questions about the practical implications of the research. All of this served to give this dissertation more focus. Financial grant for the CANDIES project from The Measurement and Control Engineering Center (MCEC) at the University of Tennessee is gratefully acknowledged. The support and opportunities provided by the Chemical Engineering Department at The University of Tennessee, Knoxville are also appreciated. Thanks are due to Charles E. A. Finney for providing several symbolization routines and the fluidized bed data sets, and to Carl R. Menako who collected some of the data sets used in this study.

Table of Contents

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Nomenclature

δ_i	Perpendicular Distance from the Principal Curve
ε	Radius of neighborhood (in correlation sum)
λ_i	Arc length (for a point)
μ	Mean (of univariate PDF)
Θ	Kernel Function
σ	Standard Deviation (of univariate PDF)
$\boldsymbol{\Sigma}$	Diagonal Matrix containing eigenvalues
τ	Embedding Delay or Lag; symbolization interval
ω_{i}	Eigenvector
ξ_i	Eigenvalue
A	Parameter in Logistic Map
$\mathbf{C}$	Variance-Covariance or Correlation Coefficient Matrix
C^{m}	Correlation sum
D	Dimension
$\mathbf{f_k}$	k$^{\mathrm{th}}$ cluster center in CLPC
F	Joint probability distribution function
$\mathrm{H_i}$	Shannon Entropy
$\mathrm{H_S}$	Modified Shannon Entropy
I_i	Redundancy, Mutual Information
I_τ	Mutual Information for a time series at lag τ
J_k	Discriminatory power of feature k
m	Embedding Dimension; Symbol Sequence length
n_c	Number of clusters (in CLPC algorithm)
$p_{i,j}$	joint probability

q	Parameter in generalized correlation sum
s	Symbol set size
$\mathbf{T}$	Delay operator (in mutual information)
T	Theiler Correction in correlation sum
u_i	i^{th} Eigenvector or Singular Vector
$\mathbf{U}$	Principal Component Matrix or Loading Matrix
x_i	Individual Measurements
$\mathbf{x}_{\mathrm{i}}$	Multidimensional vector (or embedding vector)
X,Y	Probability Space

Glossary of Acronyms

CLPC	Cluster-linked principal curve
HSPC	Hastie and Stuetzle's Principal Curve
IWLS	Iterated weighted least squares
K-S Test	Kolmogorov-Smirnov test
LDA	Linear Discriminant Analysis
MIF	Mutual Information function
NLPCA	Nonlinear Principal Component Analysis
PCA	Principal Component Analysis
PCOP	Principal Curve of oriented points
SPC	Statistical Process Control

Prediction is very difficult, especially of the future
-Niels Bohr

Chapter 1

Background and objective

1.1. Introduction

It has always been mankind's quest to precisely predict the future outcomes of event(s) and evolution of varying or dynamic things. The sunspot activity, for example, was studied as early as 37 BC [Needham (1959)].

There are two aspects to understanding dynamic processes. If a certain phenomenon is not too complex, one can learn how it normally behaves. This aspect of recognizing patterns or salient features was central to the survival of humankind. The shift from hunting and gathering to farming was brought about precisely because the ancient man recognized the regularity of seasons and exploited it to produce a reasonable harvest under many uncertainties. The uncertainties in their turn spawned a plethora of rituals, with the goal of accounting or compensating for the unaccountable or unknown.

One the other hand, the ability to detect or predict a change or shift has also been very crucial for the survival of humankind. Considering the example further, the ancient man had to understand that the land had become barren or arid and would not bear crops anymore. That is an example of detecting a change in the existing, regular, or normal behavior.

The two concepts are related, for, if there is no normal behavior, or the normal behavior lasts only for short intervals (in which case it is not really normal), it becomes extremely difficult to make informed decisions. What happened to the ancient man under rapidly changing conditions around him is hard to say (the great ice ages come to mind whereupon man probably migrated to warmer climes), but the dinosaurs, for example, vanished from the face of the earth presumably because they could not adjust to the drastic changes in earth's climate (it is quite likely they recognized the change after it had long since settled in, but were helpless to do something about it, although they had existed for millions of years).

In earlier times, any kind of change that could not be explained was attributed to gods, dæmons, totems or spirits. The work of the ancient Greeks and then that of Newton promoted the powerful argument that the physical systems were indeed predictable, since they are governed by certain rational laws. Newton said that if god created the universe, it had to be beautiful, in keeping with the spirit of perfection that was the *zeitgeist* of the age of Enlightenment. It took several centuries before Heisenberg with his uncertainty principle cast serious doubts about a Newtonian Universe.

In the Newtonian world (which is still the central paradigm for engineering science), all that one needs to understand a system or phenomenon are the deterministic equations or laws governing the dynamics of the system under study. Dynamics is typically concerned with change or movement taking place over time.

In order to have a better estimate of the change over time, some measurements need to be made on the system of interest. Thinking probabilistically, these measurements are random samplings from a *finite set*. The underlying assumption is that measurements are generated by and provide information about a generating process, which may or may not be visible. The generating process can be a physical, chemical or biological system, about which much or little may be known. Associated with a generating process is a conditional probability –which imposes certain restrictions on the sampling process –and hence the measured properties.

In the quest to explain the structure present in the measurements, the simplest approach, of course, is to model a system with difference or differential equations that utilize our knowledge of the physics, chemistry or biology. However, in many cases the underlying theory is scant, and one is presented with not much more than the data itself. In such cases the goals are to recognize important features or patterns in the data set, and to have a way of approximating the behavior of the system.

Many phenomena in our environment are studied using sequences of measurements or observations, made over time. These sequences of observations, called time series, often comprise an important part (and in some cases the only source) of the information available on the system being studied.

The analysis of time series has broad applicability over otherwise disparate fields of research. In the engineering community the term signal is used more often. However, the term time series is more generic, and applies to discrete or continuous measurements. Also, in some cases, the measurements are made not over time, but over some other variable, for example the length. This study pertains to the sequences where the measurements are made over time –and hence the term time series is more appropriate.

It is suitable here to introduce the basic notation. x_t or $x(t)$ is a measurement made at time instant t about the property that is represented by x. $\{x_t\}$ or $\{x(t)\}$ is the set of the measurements and a shorthand notation for $\{x(t)|t=0, 1, 2,...\}$.

This study concerns itself with characterizing a system, and to detect a change in its dynamics or in the underlying generating process as soon as possible. The findings of this study and the methods expounded are very practical, and can be used profitably for monitoring, fault diagnosis and control. It is assumed that the data comprises multiple measurements made over time. Albeit it is preferable to exploit the understanding of the system to the largest possible extent, the results of this study apply even when the data is the only source of information.

The layout of this dissertation is as follows. This chapter provides a short review of modeling concepts. Chapter 2 outlines basic concepts of nonlinear dynamics and chaos, and introduces the different approaches taken for characterizing them. Chapter 3 furnishes a brief description of the experiments which serve as data sources for this dissertation. Chapter 4 develops, discusses and demonstrates the cluster-linked principal curve (CLPC) algorithm. Chapter 5 deals with the uses of Cluster-linked Principal Curves in testing for stationarity and reversibility, and in comparing different processes based on time-based return maps or delay space embeddings. Chapter 6 discusses some information theoretic measures of a time series. Chapter 7 concerns itself with examples of how information theoretic measures can be used to characterize a system and to compare different processes. Chapter 8 contains the conclusions and suggestions for future directions of research.

1.2 System modeling and identification

System Identification is the discipline of making mathematical models of systems from experimental data, measurements or observations. The goal of modeling or identification is to capture the essential features of the observed patterns in the system behavior and to increase our understanding of the generating process, or dynamics, of the observed system.

For most natural processes, the measurements are influenced by some *random mechanism* no mathematical model can adequately describe. Even when an exact mathematical solution for a system exists, there are some unavoidable measurement errors, and these errors by their very nature, are random quantities. There are two kinds of models –deterministic and stochastic. The formal definitions for deterministic and stochastic models are presented now.

1.2.1. Deterministic and stochastic models

In some cases, it is possible to derive a model based on physical laws, and thus calculate some time-dependent quantity nearly exactly *at any time*. Such a model is completely *deterministic*. In that case, it is possible to write a mathematical equation such as:

$$x(t)=f(t) \qquad (1)$$

Where f is a function defined for all t such that $f(t)$ is always finite.■

However, unless there is a complete understanding of the theory describing the process, there will always be unknown factors at work. Besides, there are measurement errors associated with real processes. The uncertainty –caused by dynamic and white noise –may preclude a precise and exact deterministic model.

Nevertheless, it is possible to predict that a future value should reside within a range. Such a model is called a *probability model* or a *stochastic model*. This could be represented as

$$x_i=f(x_{i-1},\ x_{i-2},\ldots,\ x_{i-m})+e_i \qquad (2)$$

where e_i is the 'noise' whose properties are unknown, and unknowable■

The basic difference is that the deterministic approach *assumes* there is a *deterministic* structure in the data that can be explained with appropriate equations[1], whereas the statistical approach treats time series measurements as random values, and assumes no structure –but exploits the correlation to estimate the model parameters.

It must be noted here that the dichotomy between deterministic and stochastic models is not rigid, since in many fields, especially in engineering, one often has a deterministic system with stochastic elements. The stochastic element may be present as white noise (in which the system parameters remain unchanged) or dynamic noise (in which the system parameters are influenced by stochastic fluctuations). It is the strength of the deterministic elements relative to the stochastic ones (Signal-to-Noise ratio in Electrical Engineering community), and the domain of the latter that dictates the preferred modeling approach.

[1] It means that knowing the value of a variable at any one time (initial condition) allows one to calculate the value of that variable at any given instant of time

Modeling can be parametric, or non-parametric. Parametric *does not* mean the absence of parameters, but the absence of any assumptions about the distribution of observations. For example, consider the autoregressive model:

$$x_t=\beta_1 x_{t-1}+\beta_2 x_{t-2}+\ldots+\beta_m x_{t-m}+e_t \tag{3}$$

The coefficients $\{\beta_i\}$ in the equation can be estimated by Multiple Linear Regression (MLR), but in order to quantify the uncertainty in these parameters, MLR assumes that the residuals or the model errors (e_i in equation 3) are normally and independently (meaning no correlation) distributed. Linear regression also assumes that the regressors (x_{t-1}, x_{t-2}, etc in equation 3) have no measurement errors. If the assumptions are invalid, the model may be a poor estimate of true dynamics despite there being a linear relationship as described in equation (3). In this study, no assumptions are made about the probability distribution of the observations.

1.2.2. A definition of stationarity

It was mentioned before that the generating process uniquely identifies the state of the system. The invariance of the generating process is called *stationarity*. Stationarity means the underlying process generating the measurements does not change over time. The invariance of the generating process can be formulated as the invariance of the joint probability distribution.

A precise asymptotic definition exists [Diks (1999)], as shown below.

A bounded, infinitely long time series $\{x_1, x_2, ... x_n\}$
is considered to be stationary if the averages

$$\bar{g} = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} g(x_k, x_{k+1}, ..., x_{k+m\text{-}1}) \tag{4}$$

exist for each m and each continuous function g: $R^m \to R$ ■

Under this condition, there exists an associated *probability measure*, called the *reconstruction measure*. Some functions that can be considered are the moments –viz., mean, variance, and higher level moments like kurtosis and skewness.

Another definition given in Tong (1990) follows.
A time series $\{x_1, x_2, ... x_n\}$ is said to be stationary if,
for any $t_1,\ t_2,\ ..., t_n \in Z$, any $k \in Z$, and $n = 1,2,...$

$$F_{x_{t_1}, x_{t_2}, ..., x_{t_n}}(x_1, ... x_n) = F_{x_{t_1+k}, x_{t_2+k}, ..., x_{t_n+k}}(x_1, ... x_n) \tag{5}$$

where F denotes the (joint) probability distribution function of the set of random variables that appear as suffixes. ■

It must be recalled here that comparing probability distributions is quite subjective, if one chooses to forego rigid assumptions about them. *Bootstrapping* methods, while very useful, are hard to implement with time series data because of the temporal correlation present in the latter. One cannot randomly generate sub-samples without losing much useful information in the time series.

1.2.3. Linear systems theory

1.2.3.1. The correlation approach

Generally, the observations at time instant *t* are correlated with those at *t-2, t-1, t+1, t+2,* etc. Such a property of a time series is called its *autocorrelation*. A definition of autocorrelation is given below.

$$\rho(r) = \frac{\sum x(t)x(t+r)}{\sum x(t)^2} \tag{6}$$

where $\rho(r)$ is the autocorrelation of series $\{x_t\}$ for lag *r*. The index *t* covers all the records in the time series.■

Statistical linear methods like ARIMA (Auto-Regressive Integrated Moving Average) models exploit the temporal autocorrelation to provide a *parametric* model for a given time series. Linear Model Building attempts to fit a model to a time series with a minimum number of parameters, so that the *residuals* or the *model errors* have an *I.I.D.* (independently identically distributed) probability distribution[2].

If the observations (to be predicted) are not stationary, then the linear systems theory is not relevant. One can attempt to monotonically transform the variable to achieve normality. Such methods include, among others, Box-Cox transformation and taking logarithms.

[2] Requirements of I.I.D. process amounts to null autocorrelations and a normal running PDF for any lag.

The Auto-regressive (AR), Moving Average (MA), and Auto-regressive Moving Average (ARMA) equations can also be written as difference equations, and can represent the input-output relationship as a Z-transform. That approach is often preferred in the engineering community.

1.2.3.2. The spectral approach

Another way to look at linear models is by considering *Fourier analysis*, in which the time series is modeled by a weighted sum of orthonormal sinusoids, thus establishing a one-to-one mapping between frequency- and time- domains. A definition of discrete Fourier transform (DFT) is given here. There are many definitions, but the one provided has the advantage of having similar-looking expressions for DFT and Inverse DFT (IDFT).

The Fourier transform of a time series $\{x_t\}$ can be defined as:

$$\mathrm{h}(\omega)=\frac{1}{\sqrt{2\pi}}\sum_{t=0} x_t e^{-j\omega t} \tag{7}$$

The inverse Fourier transform is defined as

$$x_t = \frac{1}{\sqrt{2\pi}}\sum_{\omega} \mathrm{h}(\omega) e^{j\omega t} \tag{8}$$

The power spectral density (PSD) is defined as

$$P(\omega) =\|\ \mathrm{h}(\omega)\ \|^2 \tag{9}$$

Note that the discrete Fourier transform is a complex number, but the power spectral density is a real number.■

Replacing the summation by an integral yields the expression for their continuous counterparts.

It is interesting to note here that under the condition that the autocorrelations exist and are finite, the autocorrelation function uniquely determines the Fourier transform. An equation similar to equations (7) and (8) can be written to relate the Fourier transform to the autocorrelations $\rho(r)$ instead of the time series measurements $\{x_t.\}$ Theoretical details can be found in Priestley (1981).

Based on the discussion above, it is clear that the different approaches taken in linear systems analysis are essentially the same and they make certain consistent assumptions about the data. Any apparent differences are due to the different ways the topic has been approached by researchers in various fields.

The time series that can be modeled as an Auto-Regressive Integrated Moving Average (ARIMA) processes, Z-transforms or Fourier series must possess two properties, namely those of *stationarity* and *reversibility*. The concept of stationarity has already been reviewed. Reversibility means that the joint probability distributions of the time-forward and time-reverse versions of the time series are virtually indistinguishable. In other words, the essential properties of the time series and its time-reversed version are the same, and there is no *arrow of time* or entropy-maximization at work.

A key objective of this study is to find ways to characterize a time series, and to determine if a time series is stationary and reversible. Standard time series analysis and prediction tools are useful in their own right for stationary and reversible time series, but not otherwise. Hence the results of this study are relevant even for linear processes.

Chapter 2 introduces some basic concepts of nonlinear dynamics and chaos and shows why linear methods outlined in this chapter cannot be effectively used for nonlinear systems.

Wo das Chaos auf die Ordnung trifft, gewinnt meist das Chaos,
weil es besser organisiert ist.
-Friedrich Nietzsche

We adore chaos because we love to produce order.
-M. C. Escher

Chapter 2

Nonlinear dynamics and chaos

Physical systems can be described by equations governing their evolution. *Dynamics* is the study of such equations. A linear system is a system whose time evolution equations are linear, that is, the state equations describing the system can be written as a linear combination of the variables describing system properties. The state equations of nonlinear systems do not permit linear decomposition, which is why they are inherently more difficult to analyze. The systems whose state equations contain coefficients that do not depend on time are called time-invariant. Linear Time Invariant systems have been studied in great detail, and their theory stands perfected with mathematical simplicity and elegance.

Mathematically, linear systems must meet the constraints of superposition principle. What that means is essentially the whole is the sum of the parts, and which can be written as:

$$f(\alpha x_1+\beta x_2)=\alpha f(x_1)+\beta f(x_2) \tag{10}$$

where x_1 and x_2 are variables and α and β are constants. ■

2.1. An example: the logistic map

A simple example of a nonlinear system is the logistic map, described by the simple equation

$$x_{n+1}=Ax_n(1-x_n) \text{ where } x_i\in[0,1]. \tag{11}$$

The subscript refers to the time index. A ($0<A<4$) is a parameter■

The logistic map is a simple model for the evolution of a biological population size x of some species from generation to generation. When the population is low, the relative abundance of resources results in fast population growth. However, the increased population causes more competition for the available resources, or their exhaustion that leads to a reduction in population. The model, admittedly a very simple one, has very interesting behavior unexpected from such an innocuous equation. More information can be found in an influential article [May (1976)] published in Nature.

Figure 2.1 shows the time series for A=3.9 and starting values x_0 of 0.7499 (dashed line) and 0.7500 (solid line). Note the high sensitivity to initial conditions in Figure 2.1. The evolution of the time series with nearly the same origin tracks each other only for a few iterations, and after that the time series exhibit no relationship to each other.

2.2. Terminology

It will be useful to introduce some terminology here. Consider that the vector $\mathbf{x}_i$ completely and unique describes the system at any time instant i.

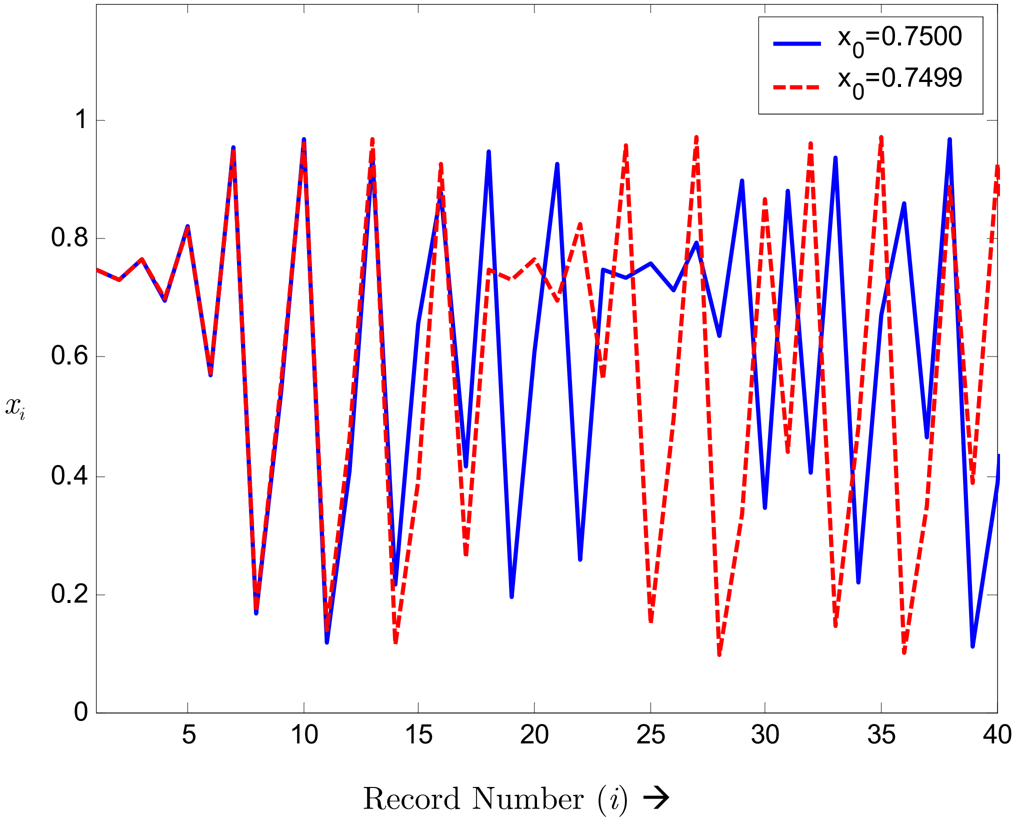


Figure 2.1. Two time series obtained from the logistic map.

The solid time series corresponds to A=3.9 and x_0 =0.7500, whereas the dashed line corresponds to A=3.9 and x_0 =0.7499. The series track each other for about 12 iterations, and then begin to diverge visibly. The Lyapunov exponent for the logistic map at A=3.9 is roughly 0.63. The Lyapunov exponent is a measure of how the natural logarithm of the distance between two trajectories with very similar initial conditions diverges. In this case, the distance between the trajectories started off from x_0 =0.7499 and x_0 =0.7500 grows by a factor of $e^{0.63}$ or 1.878 after every iteration. See equation (11) for the time-evolution of logistic map.

Assume that the vector is made up of individual measurements or observations $\{x_{1i,}\ x_{2i,}\ x_{3i},\ldots,x_{mi}\}$, where the arabic numerals index the measurements. If a vector unequivocally characterizes a system, it is called the *state vector*. The space R^m where the state vectors reside is called the *state space*. Normally the dimension of a state space is the minimum number of variables needed to uniquely characterize the system, or its degrees of freedom.

Consider a two-dimensional state space. In that state space, the sequential set $\{(x_1,\ y_1),\ (x_2,\ y_2),\ldots,\ (x_n,\ y_n)\}$ or $\{\mathbf{x}_1,\ \mathbf{x}_{2,}\ \ldots,\ \mathbf{x}_\mathbf{n}\}$ describes the path traversed by the system from time instants 1 through n. That path is called a *trajectory*. The extension to higher dimensions is straightforward. Sometimes the term *phase space* is used interchangeably with state space. However, the term phase space was introduced by Gibbs in relation to thermodynamics, and is much more restrictive. The phrase state space is used in this dissertation. The term pseudo state space pertains to a state space that does not necessarily relate to the physical properties of the system. For the sequential set described in this paragraph, replacing y_i with x_{i+1} furnishes an example of *pseudo state vector* and *pseudo state space*.

All these trajectories can be thought of as a *mapping*. That is to say that $\mathbf{x}_1$ maps to $\mathbf{x}_2$, and the mapping function is, say, f(**x**). It is not necessary for f to have a closed mathematical expression. All that is required is that the mapping be unique: invertibility is not necessary.

Stable linear systems are stable in the BIBO (Bounded Input Bounded Output) sense. It means that nearby trajectories remain close in the state space at all times. This is not true of nonlinear systems. Chaotic systems are the extreme example of nonlinear systems and are characterized by exponential divergence of nearby trajectories.

The exponential divergence of nearby trajectories has been immortalized by the picturesque *butterfly effect,* attributed to Lorenz. He had developed a simple model to describe the atmosphere, and his equations demonstrated sensitive dependence on initial conditions or exponential divergence of nearby trajectories. Lorenz had in fact used a seagull as the metaphor to verbalize the finding that *the flapping of a seagull's (or butterfly's) wings could render long-term prediction of weather useless*.

The sensitive dependence on initial conditions rules out long-term prediction, since even a small error in a measurement or even prediction (due to limited storage for a computed value), grows exponentially over time so much so that after some period of time, the predictions would be utterly inaccurate. For chaotic systems, prediction can only be done for short term. However, that is not necessarily a liability.

The chaotic systems considered so far had a set of differential or difference equations that described their evolution. In other words, the systems were deterministic in the sense that one knows how the system will behave given some information about how it behaved in recent past. The term chaos has been used in various contexts, most of whom stem from mythology. What is meant by chaos here is deterministic chaos, i.e., the complex behavior of systems that otherwise follow physical laws, and are not random.

The term *fixed point* is used to describe the solution of the governing equation (for maps[3]), or the point where all derivatives (recall the mapping function f) are zero (for flows[3]). It may be *stable*, *unstable*, or a *saddle*

[3] If the governing equation is a difference equation, it is called a map. And if it is a differential equation, it is called a flow.

point, in short, a point whither the trajectories are *repelled* or *attracted*. For a stable fixed point, the trajectory terminates upon reaching it. An unstable fixed point, unlike the fixed point, repels any trajectory reaching it. For a saddle point, there are some directions along which the trajectories are attracted, and some directions along which they are repelled. A trajectory cannot stay at the saddle point for long: it is soon repelled from it along the repelling directions.

An *attractor* is a set of such *attracting* points, and can be a point, a line, a curve, or a surface; it is an attracting set all trajectories starting in the *basin of attraction* eventually reach (given enough time). Note that this study is concerned primarily with dissipative systems, i.e., systems that are characterized by a shrinking volume in the state space or negative divergence. Most physical systems have only finite energy, which is slow lost or dissipated due to friction or other such effects. This study focuses on dissipative systems.

Dissipative chaotic systems are characterized by *strange attractors* having fine, layered, *fractal* structure produced by folding and unfolding, or kneading and stretching of a map. For such systems, given enough time, any trajectory, winding through the state space, though infinitely long, occupies zero volume. For illustration, please refer to figure 2.5 that shows the pseudo state space for Rössler equation.

2.3. Poincaré sections and return maps

The divergence of nearby trajectories has been known for over a century. French mathematician Henri Poincaré was the first to remark at the basic concepts of nonlinear dynamics and chaos. He noticed the phenomenon in his research on the behavior of several planets interacting with each other.

Return Maps are very instructive in understanding nonlinear structure in the data. In Return Maps, the measurements at any time $\{x_{t+q}\}$ are plotted against their lagged counterparts $\{x_t\}$. Figure 2.2 shows the return map for the logistic map time series with A=3.9 and x_0=0.75.

The dots are data points, and the diagonal is the 45-degree line on which the fixed point resides. The asymmetric distribution of the points about the diagonal reveals that the behavior of the logistic map is not symmetric in time. Such time-asymmetry is in fact a characteristic of most nonlinear systems.

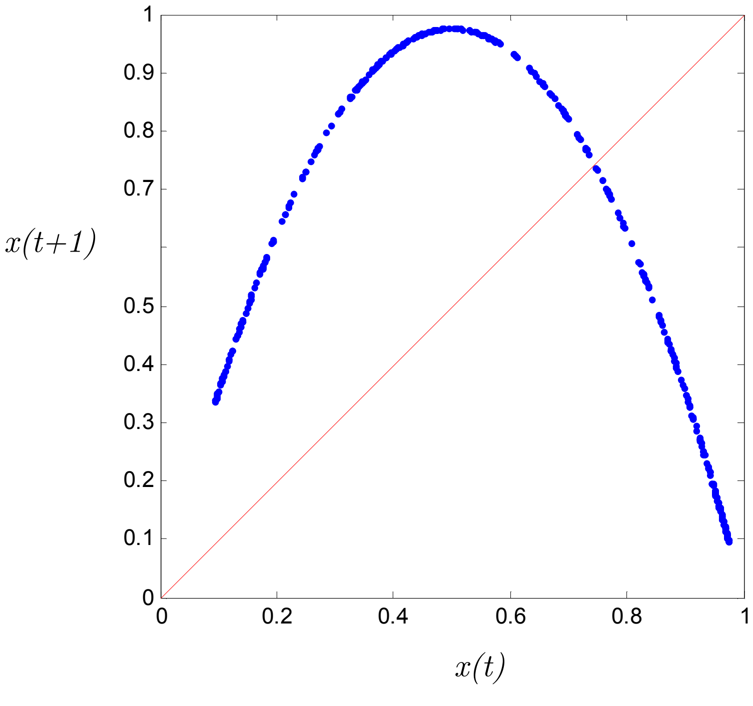


Figure 2.2. Return map for the logistic map time series (delay =1).

See equation (11) for the logistic map equation. For this figure A=3.9

Figure 2.3 shows the return map of the same time series, but with delay of 10. The clear pattern in figure 2.1 is lost, and the data points are apparently random. In short, one has short prediction horizons, which is another hallmark of chaos.

Poincaré also introduced a simple but useful concept. Imagine a plane across the state space. *Poincaré sections* are the points on that plane where the trajectory pierces the plane in any one chosen direction (transverse crossings only need be considered). Let us call the point where the trajectory cuts the plane z_i. Note that one has to define the direction of crossing (to differentiate between the trajectories going from left to right and right to left). The procedure will provide a sequence of such numbers. If one has the equations for the system, it is theoretically possible to find a function that could predict the next crossing given the current crossing. For example, a

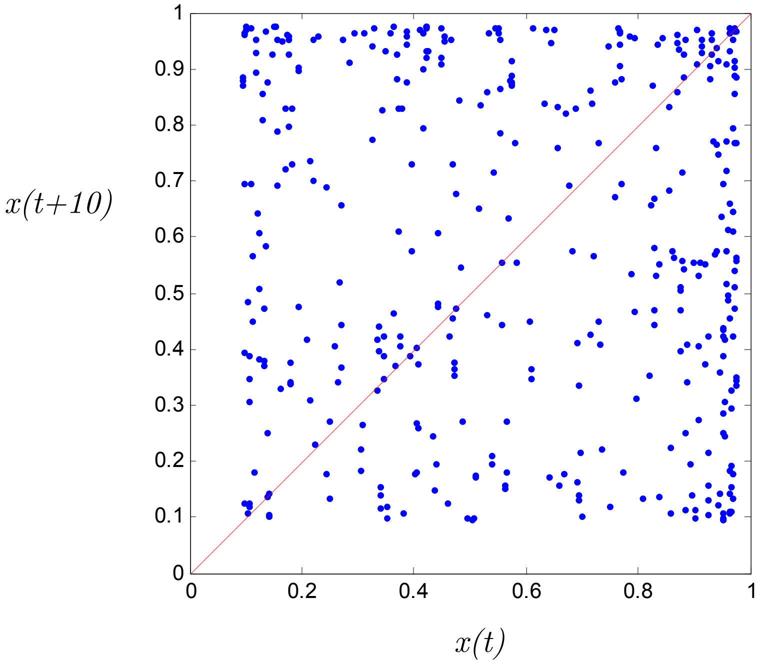


Figure 2.3. Return map for logistic map time series (delay=10)

See equation (11) for the logistic map equation. For this figure A=3.9

two-dimensional pseudo state space could be constructed, and the ordered sequential set $\{(z_1, z_2),(z_2, z_3),\ldots,(z_n, z_{n+1})\}$. The resulting plot is called Poincaré map. Poincaré maps could also be constructed of measurements recorded at fixed time intervals.

Poincaré maps serve to reduce the dimension of the state space by unity. An approach used by Nguyen et al (1996), considers the time intervals between successive piercings of the plane (while obtaining its Poincaré sections), in plotting return maps. A return map based on the mean crossings is called *time-based return map*. Figure 2.4 shows the time-based return map of the logistic map time series.

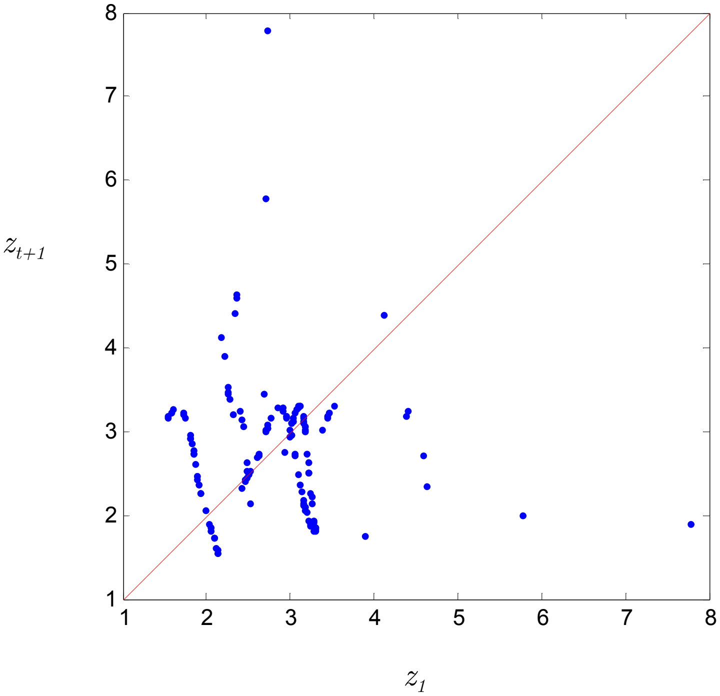


Figure 2.4 Time-based return map for the logistic map time series

For this figure A=3.9

Without additional information, a sensible choice for the cutting plane is the mean of the time series. In plotting figure 2.4, the mean of the time series was used as the cutting line in order to obtain the *mean crossings*.

Figures 2.2 through 2.4 use the same data. One can observe how, for most part, the mean is crossed every 2 to 4 time periods in figure 2.4. Sometimes, though, the time series stays above or below the mean for longer periods. By changing the level for determining crossings, more information can be had about the time series at hand.

The logistic map data is not a good candidate for time-based return maps. Time-based return maps are quite useful about the physical systems that result in time series with some sort of periodicity. They are not much useful for a discrete equation like the logistic map equation. From this point on, time-based return maps are referred to as return maps. Return map in its original definition is called delay space.

2.4. Embedding and pseudo state space

It was shown by Sauer et al (1983), that a system can be adequately represented by its *embedding vectors*. Embedding vectors are formed by treating time-lagged measurements as co-ordinates in the *reconstructed delay-space* or *pseudo state-space*. They found that even for a system with many degrees of freedom (or measurable variables), information on only one variable, if collected sufficiently well, is enough to reproduce the geometry of the attractor.

Assume that all values in one time series, $\{x_i\}$ are drawn from a probability space. Let this probability space be X_0, which is the collection of all possible values in the time series. The embedding vector can be defined as follows

$$\mathbf{x_i} = \begin{bmatrix} x_i \\ x_{i+1\tau} \\ x_{i+2\tau} \\ \cdot \\ \cdot \\ x_{i+(m\text{-}2)\tau} \\ x_{i+(m\text{-}1)\tau} \end{bmatrix} \qquad (12)$$

where

$\mathbf{x_i}$ = Embedding Vector
x_i = Individual measurements.
m = Embedding Dimension
τ = Embedding Delay ■

Considering X as the *m*-times Cartesian product of X_0 with itself, it is the probability space where the embedding vectors reside. Thus $\mathbf{x_i} \subset X$ and $\mathbf{x_i} \in R^m$. Abusing the terminology a little, let us denote by X the probability space as well as the process that generates it.

There are two parameters involved in formation of embedding vectors –m and τ. However, what is important is the time interval in the window – i.e., *(m-1)τ*. This time interval must be large enough to resolve the dynamics. Too small an interval covers only a small part of the entire state space; too large a window of course, adds no information for systems characterized by short-term memory.

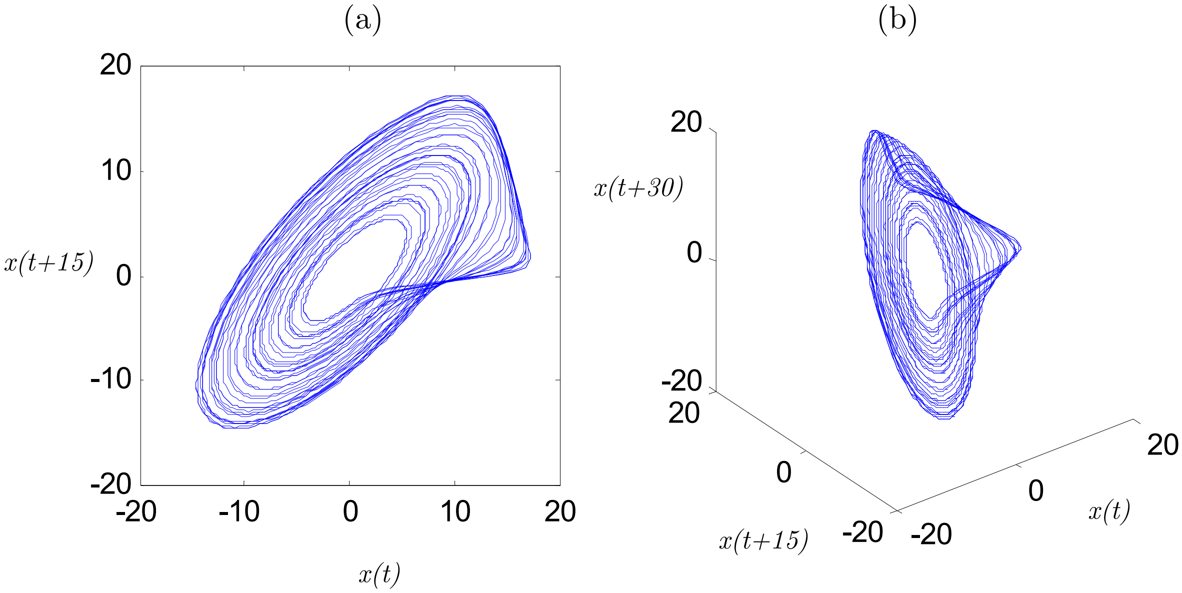
Takens (1980) established the upper limit for the embedding dimension for reconstruction of attractor geometry. If *D* is the dimension of the *true state-space*, then the maximum embedding dimension m_0 required for a faithful representation of system geometry is $m_0=2D+1$. The logic behind establishing the correct embedding dimension is that by increasing the pseudo state-space dimension, the self-crossings of the orbit, caused by the projection of state space to a too low a dimension will be eliminated.

Let us consider the Rössler system described by the following equations. For details, see Rössler (1976).

$$\begin{aligned} \frac{dx}{dt} &= -y-z \\ \frac{dy}{dt} &= x+ay \\ \frac{dz}{dt} &= b+z(x-c) \end{aligned} \tag{13}$$

Figure 2.5 shows the embedding of the *x*-component of the Rössler equations. We used *a*=*b*=0.2 and *c*=4.7 –which is a well-studied case for these equations. The time series was obtained by prescribing random initial conditions and integrating with a fourth-order Runge-Kutta method. The first 10 seconds were discarded, and the following data was used. It can be seen that the apparent self-crossings in the 2-D plot (figure 2.5 (a)) are not reflected in the 3-D plot (figure 2.5 (b)). The dimension of Rössler attractor is between 2 and 3 –which means that the embedding dimension has to be greater than 2 to eliminate the self-intersection of trajectories.

Takens' theorem provides only an upper bound for the correct embedding dimension. It is possible to reconstruct the geometry with smaller embedding dimension. As seen in the figure 2.5, the reconstruction was quite good for *m*=3 (cf. m_0=2.5*2+1=7). Discussion regarding the smallest



Embedding dimension =2

Embedding Dimension=3

Figure 2.5. Rössler time series in embedding space

sufficient embedding dimension can be found in Kennel et al (1992). The issue of ascertaining the dimension of an attractor is taken up in section 2.5.

Henceforth embedding vector signifies the embedding vectors formed from a single, scalar time series. In case of multiple time series or a vector time series, the embedding vector can be formed considering all the components simultaneously. For example, the embedding vectors formed by individual time series can be concatenated to form a composite embedding vector. However, one should take care to consider the same time window in the entire embedding vector. All the results and comments about the embedding vectors that follow apply equally well to embedding vectors formed from multiple time series.

2.5. Characterizing nonlinear dynamics and chaos

In this section, the measures that define a nonlinear system are considered. Recalling the discussion in Chapter 1, the unique way of characterizing a system is by its joint probability distribution function. However, it may not be possible to measure all the system properties. In that case, one relies on embedding vectors to provide a good estimate. Therefore, the discussion is based on embedding vectors.

2.5.1. Geometrical measures

A simple way of quantifying the distribution of a set of state space points system is the *correlation sum* introduced by Grassberger and Poccacia (1983). The definition of correlation sum is now introduced.

Correlation sum $C^{m}(X,X,\varepsilon)$ is defined as follows:

$$C^{m}(X,X,\varepsilon) = \frac{1}{\binom{N}{2}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Theta(\varepsilon - \| \mathrm{x_i} - \mathrm{x_j} \|) \tag{14}$$

where the superscript m refers to the (embedding) dimension of $\mathrm{x_i}$ and $\mathrm{x_j}$. $\mathrm{x_i}$, $\mathrm{x_j} \in R^{m}$ and $\mathrm{x_i}$, $\mathrm{x_j} \subset X$. Θ is a kernel function usually taken as the heaviside step function or a radial basis function. ε is a parameter related to the partition of the space. Two points less than ε apart are considered ‘un-different’.■

The search for the nearest neighbors is carried out in a hypersphere of radius ε. The more pairs closer than ε, the higher the correlation sum. Note that the choice of ε is not arbitrary, since too small an ε will result in very low correlation sum, whereas too large an ε yields a correlation sum near unity.

Because most time series are heavily correlated, one must be careful to avoid a biased sample. To deal with this, Theiler (1986) suggested excluding measurements less that T time intervals apart from the computation of correlation sum. They argue that *temporally close* vectors are highly likely to be *spatially* and *dynamically close* and hence add no new information. Schreiber and Kantz (1997) have suggested that the measurements that are dynamically *too close* also be excluded from consideration while computing the correlation sum. Theiler's method, though, is easier to implement.

Correlation sum with Theiler correction is defined as follows:

$$\mathrm{C}^{\mathrm{m}}(\mathrm{X},\mathrm{X},\varepsilon) = \frac{1}{\begin{pmatrix} N-T+1 \\ 2 \end{pmatrix}} \sum_{i=1}^{N-T} \sum_{j=i+T}^{N} \Theta(\varepsilon - ||\mathrm{x_i} - \mathrm{x_j}||) \tag{15}$$

where $T \leq 1$ is the Theiler correction■

The correlation sum depends on embedding parameters, i.e., embedding dimension m, embedding interval τ, and the radius of the neighborhood ε, as well as the kernel function Θ. The norm $||.||$ can be taken as a Euclidean or the *sup* norm. If the *sup* norm is chosen, the hypersphere essentially becomes a hyperprismoid. With $T=1$, equation (15) reduces to equation (14).

Similarly, the cross-correlation sum $\mathrm{C}^{\mathrm{m}}(\mathrm{X},\mathrm{Y},\varepsilon)$ can be defined as

$$\mathrm{C}^{\mathrm{m}}(\mathrm{X},\mathrm{Y},\varepsilon) = \frac{2}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \Theta(\varepsilon - ||\mathrm{x_i} - \mathrm{y_j}||) \tag{16}$$

where $\mathrm{x_i}, \mathrm{y_j} \in \mathrm{R}^{\mathrm{m}}$ and $\mathrm{x_i} \subset \mathrm{X}, \mathrm{y_j} \subset \mathrm{Y}$ ■

Note that in equation (16), the vectors $\mathbf{x}_i$ and $\mathbf{y}_j$ are formed from different measurements (X and Y respectively). The divergence of $C^{m}(X,Y,\varepsilon)$ with increasing partition can be quantified as the correlation dimension. Correlation sum is also related to Shannon Entropy and Renyi Entropies in Grassberger et al (1991).

For sufficiently large samples, the correlation sum scales as:

$$C^{m}(\varepsilon)=\varepsilon^{-D}e^{-mH\tau} \tag{17}$$

where D and H are the corresponding dimension and the entropy, respectively ■

Other generalizations of correlation sum account for the non-uniform density of points in the state-space. Definition of generalized correlation sum is given in Pawelzik and Schuster (1987) as:

$$C_{q}^{m}(X,Y,\varepsilon)=\frac{2}{N_1N_2}\left(\sum_{i=1}^{N_1}\left(\sum_{j=1}^{N_2}\Theta(\varepsilon-||\mathbf{x}_i-\mathbf{y}_j||)\right)^{q-1}\right)^{\frac{1}{q-1}} \tag{18}$$

where $-\infty\leq q\leq\infty$■

q=2 gives rise to the normal correlation sum introduced in equation (14). For q=1, the correlation sum yields information dimension and information entropy. A spectrum of dimensions is produced[4] over q. Another way to deal with the non-uniformity of data points is to scale them using a

[4] If the dimension depends on q, the attractor is a polyfractal, otherwise it is a monofractal.

static transform. However, one may have to decide upon the appropriate transform(s) for a given data set.

2.5.2. Information-theoretical measures

Correlation sums are closely related to information-theoretic quantities like *redundancies* and *entropies*. The *joint entropy* of two processes (time series) X and Y is defined in equation (19).

Entropies are defined as:

$$\mathrm{H}_q(\mathrm{X},\mathrm{Y},\Theta)=\frac{1}{q\text{ - }1}\,ln\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}p_{i,j}(\mathrm{X},\mathrm{Y})^q \qquad (19)$$

where $p_{i,j}(\mathrm{X},\mathrm{Y})$ is the probability of a X-vector being in bin i, and a Y-vector being in bin j. n_1 and n_2 respectively the bins in X- and Y-spaces. The symbol **Θ** emphasizes the fact that these quantities depend on the embedding parameters **Θ**■

Before the expression in equation (19) may be computed, the *corresponding* Y-vector has to be defined. If the pairs are chosen so that $\mathbf{x}_\mathrm{i}$ and $\mathbf{y}_\mathrm{j}$ are such that:

$$\mathbf{x}_\mathrm{i}=\begin{bmatrix} x_i \\ x_{i+1\tau} \\ x_{i+2\tau} \\ \cdot \\ \cdot \\ x_{i+(m-2)\tau} \\ x_{i+(m-1)\tau} \end{bmatrix};\quad \mathbf{y}_\mathrm{j}=\mathbf{x}_{\mathrm{i}+\theta}=\begin{bmatrix} y_{i+\theta} \\ y_{i+\theta+1\tau} \\ y_{i+\theta+2\tau} \\ \cdot \\ \cdot \\ y_{i+\theta+(m-2)\tau} \\ y_{i+\theta+(m-1)\tau} \end{bmatrix} \qquad (20)$$

then the expression in equation (19) is an estimate of the entropy in the joint probability distribution of X and Y. If the joint probability distribution is random (knowing $\mathbf{x}_\mathrm{i}$ doesn't tell us anything about $\mathbf{y}_\mathrm{i}$ or $\mathbf{y}_{\mathrm{i}+\theta}$), the entropy will be very high. However, any relationship between X and Y reduces the entropy, since there is more *order* in the joint probability distribution. The joint entropy is an estimate of *general independence* between processes X and Y. For X lagging Y by θ, it is also an estimate of how well one can predict $\mathbf{y}_\mathrm{i}$ vectors if $\mathbf{x}_{\mathrm{i}-\theta}$ is known.

Of course, if $\mathbf{x}_\mathrm{i}=\mathbf{y}_{\mathrm{i}+\theta}$, equation (19) is an estimate of the *generalized autocorrelation* in the time series X at lag θ.

for $q=1$, Using L'hôpital's rule, equation (19) reduces to

$$\mathrm{H}_1(\mathrm{X},\mathrm{Y},\Theta)= - \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} p_{i,j}(\mathrm{X},\mathrm{Y})\ln\ p_{i,j}(\mathrm{X},\mathrm{Y}) \qquad (21)$$

Which is the expression for *Shannon Entropy*. ■

If the distributions of X and Y are completely independent, then the joint entropy can be represented as the sum of individual entropies. The amount of reduction caused by considering the joint probability distribution as compared to individual entropies is an estimate of the information X adds about Y. Such additional information is called redundancy.

The Simplest form of *redundancy* –viz. *mutual information* can be defined as:

$$\mathrm{I}_\mathrm{i}(\mathrm{X};\mathrm{Y},\Theta)=\mathrm{H}_\mathrm{i}(\mathrm{X},\mathrm{X},\Theta)+ \mathrm{H}_\mathrm{i}(\mathrm{Y},\mathrm{Y},\Theta)- \mathrm{H}_\mathrm{i}(\mathrm{X},\mathrm{Y},\Theta)■ \qquad (22)$$

The generalization to higher dimensions is straightforward. For details see Fraser and Swinney (1986). Mutual information is the information added by X about Y, and vice versa. The redundancies can be considered a *general cross-correlation*, since a high value of redundancy implies relationship between X and Y. If Y is a time-lagged version of X, then it is an estimate of the *general autocorrelation* in the time series X.

This definition of mutual information is *non-directional*, because the definition of Entropy is symmetric with respect to X and Y. At times, the symbol Θ is dropped for brevity. It must be borne in mind though, that the quantities depend on the choice of embedding parameters. In other words

$$H_i(Y,X)= H_i(X,Y) \tag{23}$$

and, therefore

$$I_i(X,Y)=I_i(Y,X) \tag{24}$$

There is another related concept that relates to directional entropy. Kullback-Leibler Information quantity is a general concept that is used to discriminate between two different probability densities. For example, for two probability densities $F(X|\Theta^*)$ and $F(Y|\Theta)$, the K-L information can be defined as follows.

$$I(F(X|\Theta^*);F(Y|\Theta))=\sum F(X|\Theta^*)\log\frac{F(X|\Theta^*)}{F(Y|\Theta)} \blacksquare \tag{25}$$

The expression in equation (24) is not symmetric. It is an estimate of the distance between the two distributions; it can also be used as a measure of distance between two dynamical systems that produced those distribution functions.

According to the definition, it is possible to compare two different probability density functions formed with different embedding parameters. If the K-L information of many series is computed against some benchmark series, a series can be characterized by these distances. It is not trivial to choose the benchmark time series, however. They must otherwise meet some constraints –preferably the same as faced by all the time series, so that the K-L information only characterizes the difference in distributions and not, let's say, departure from normality. See Schreiber (2000) for an extension of K-L information to time series data.

Chapter 3

Experimental setup and sources of data

This section briefly describes the experimental setup of the systems whose output is used in this study. The systems are a *bubble column* with an electrified capillary, and a *fluidized bed*.

3.1. The bubble column

The bubble column apparatus is concerned specifically with the formation and behavior of gas bubbles in a liquid, when the gas is injected into a liquid-filled vertical column through a single gas injector nozzle at the bottom. Henceforth this column is referred to as *bubble column*. This section is paraphrased from Menako (2001). In all experiments the liquid was glycerin and the gas was pure nitrogen. Four nozzles were used. The first one (constructed of brass in the shape of a button –called button nozzle in this dissertation) had a diameter of 0.75 mm, and three other threaded capillaries with diameters of 0.02, 0.03 and 0.04 inches. These nozzles are referred to as nozzles A, B, C and D respectively. The flow rates ranged from 0 cc/min to 440 cc/min, and the range of electrostatic potential applied was from 0V through 20000V in increments of 1000V. The corresponding gas-phase Reynolds number in the nozzle ranged from 0 to a little over 100. The process variable recorded to characterize the dynamics was the differential pressure across the nozzle. This study only utilizes the data collected on nozzle A which had an internal diameter of 0.75 mm, and looked like a button in the top view.

Figure 3.1 illustrates typical behavior of bubbles formed from a submerged nozzle with no electrostatic potential across it. The corresponding pressure trace (as measured by a transducer in-line) illustrates how this variable changes through the various stages of growth and detachment. The key forces are surface tension and buoyancy. The former resists the release of bubble from the nozzle and the latter pulls the bubble off the nozzle. When the buoyancy force exceeds the surface tension, the bubble is released. The movement of the bubble up the column is also influenced by the liquid viscosity which dampens its movement. With the application of electrostatic potential across the nozzle, the electric forces come into play and 'pinch' the bubbles in order to minimize the interfacial surface area. As the electrostatic potential applied across the nozzle increases, the bubble formation becomes more rapid and more complex.

A schematic of the bubble column apparatus is given in Figure 3.2. The apparatus consisted of a square glass column attached to a base of Plexiglas; this is referred to as the bubble column. The gas pressure was regulated at the cylinder head, and again at the bench top pressure regulator prior to use. The nitrogen gas flow was controlled and metered via an arrangement of a control valve and a mass flow meter. A MAXTEK model MV-112 piezoelectric valve was used for gas flow control. A special throttling valve was employed, a Swagelok NUPRO type needle valve that followed the mass flow meter.

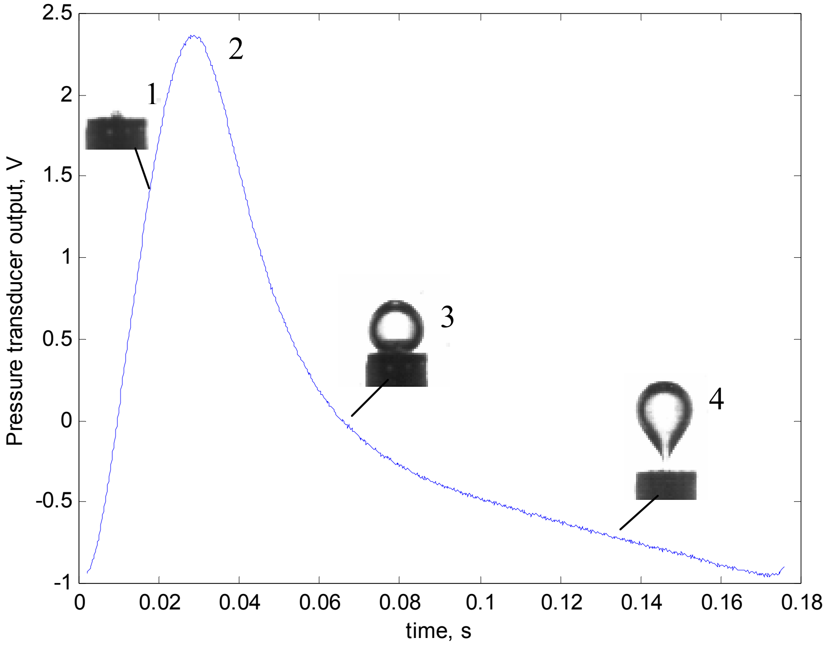


Figure 3.1: Typical pressure trace and bubble formation

Photograph of high-speed images of slow bubble formation. (1) Surface tension forces are larger than the pressure in the nozzle, preventing bubble growth; (2) Pressure in nozzle equals surface tension forces; (3) Bubble growth occurs, and (4) Buoyancy and inertial forces overcome surface tension, causing bubble detachment.

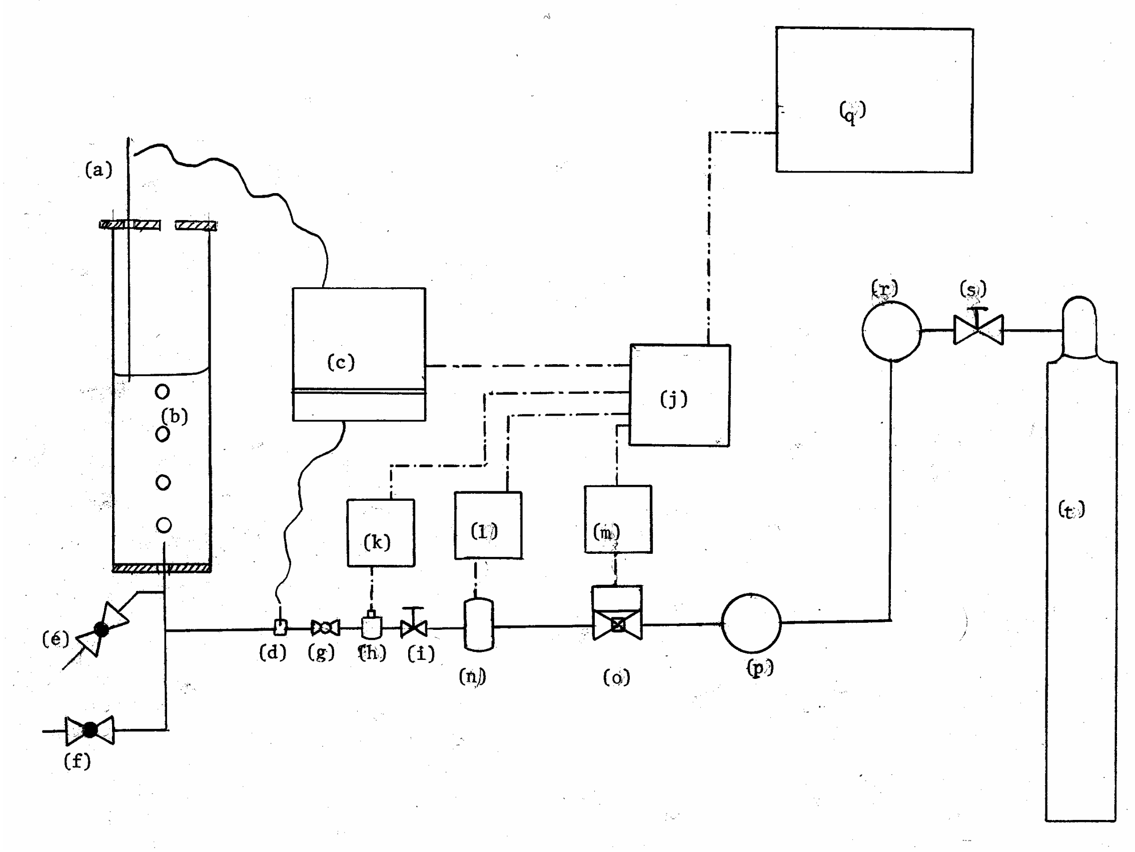


Figure 3.2 - A schematic of the bubble column apparatus.

(a) electrode (positive) polarity - 2 mm submerged into liquid (b) bubble column (c) Bertan - high voltage power supply (d) electrode (positive) polarity connected to nozzle (e) column drain (f) knock-out drain (g) block valve - ball (h) Endevco – pressure transducer (i) NUPRO type needle valve (j) National Instruments SC-2043-SG signal conditioner (k) signal conditioner (l) signal conditioner (m) signal conditioner (n) Cole-Parmer - mass flow meter (o) MAXTEK - piezoelectric control valve (p) pressure indicator (q) data acquisition system - Dell Pentium III, OptiPlex computer (r) pressure reducer (s) high pressure regulator (t) N_2 supply tank. Taken from Menako (2001).

A Cole-Parmer product, model # 32915-14 mass flow meter monitored the flow rate. The throttling valve was selected to minimize fluctuations in the gas flow-rate upstream of the pressure transducer and to stabilize the mass flow meter measurements. An Endevco® pressure transducer, model 8510B-1 was utilized to measure the differential pressure.

On the gas inlet line an interconnection to copper wire connected a high voltage power supply and the submerged nozzle. The high voltage power supply used was a Bertan Series 225. The Series 225 is a precision regulated linear power supply with a rated output voltage up to 50 kV. The Series 225 was remotely controlled via a signal conditioner. A National Instruments SC-2043-SG signal conditioner interfaced the data acquisition system. This signal conditioner interface enabled process data acquisition and control of system variables.

The time-interval between successive bubbles is employed to characterize the bubbling behavior, according to the methodology used in Nguyen et al (1996). The *bubble rate* is also used to characterize bubbling. The bubble rate is a frequency. For example, if 10 bubbles are formed in a minute, the corresponding frequency is 0.1667 Hz.

At lower gas flow rates the bubble formation is regular and almost periodic, and therefore the bubble rate remains more or less constant at a given flow rate. As the gas flow rate increases, the bubbling changes to period-2 behavior, i.e., the time-interval between bubbles alternates between a high and a low value. These low and high values do not vary much for a given flow rate. As the gas flow rate is increased more, bubble behavior changes to period-4, then to period-8, and finally becomes chaotic. A bifurcation diagram in this context is the plot of bubble rate against the gas flow rate or against the gas phase Reynolds number in the nozzle. Similarly a bifurcation diagram can be drawn for the bubble rate against the electrostatic potential. Such diagrams demonstrate how electrostatic potential causes period-bifurcation and alters the bubbling dynamics. For a more detailed treatment of the bubble column apparatus, see Menako (2001).

The bubble column is a low dimensional chaotic system that undergoes period bifurcation route to chaos. The control parameters that induce bifurcations are gas flow rate and the electrostatic potential across the nozzle. The dimension of the bubble column is between 2 and 3, but a three-dimensional embedding was found to be sufficient to faithfully reproduce the attractor geometry and to eliminate self-crossings of the trajectories.

Figure 3.3 shows the differential pressure time series for four different operating conditions –namely electrostatic potentials of 0, 12, 15 and 19 kV. The gas flow rate was 170 cc/min in all cases. All series contain 1500 records. The period-2 behavior at 0 V potential yields to a period-4 behavior at 12 kV potential, and to a possible period-8 behavior at 15 kV. Chaotic behavior is observed for the potential of 19 kV.

Figure 3.4 shows the overlaid time-based return maps for the button nozzle. The flow rate was held constant at 170 cc/min. The legend on the right refers to the electrostatic potential across the nozzle. Clearly, increasing the potential causes the period-2 behavior to give way to period-4 behavior and finally leads to chaos. With increasing potential, the bubble rate increases (the inter-bubble interval falls) but the complexity of dynamics grows from a relatively clean and simple period-2 behavior to full-fledged chaos. This behavior was typical of all nozzles.

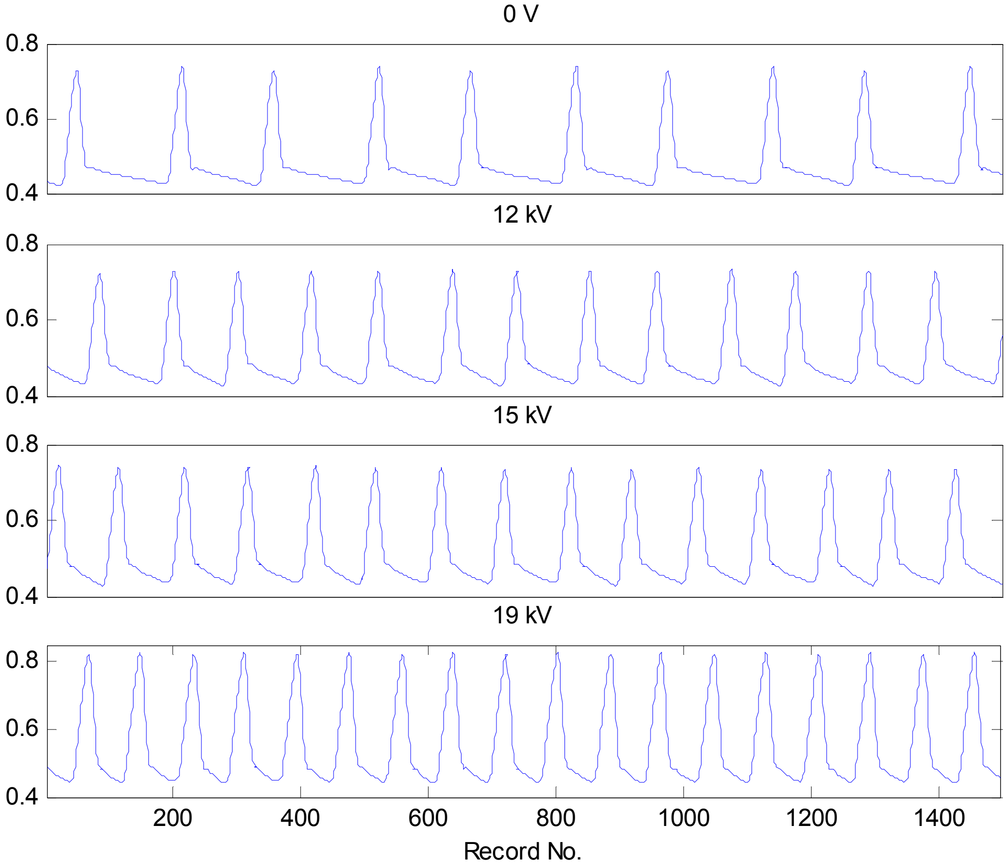


Figure 3.3. Differential pressure time series from the bubble column

From top to bottom, the series correspond to electrostatic potentials of 0, 12, 15 and 19 kV respectively. The gas flow rate was 170 cc/min for all four examples. The abscissa units are differential pressure in mm water.

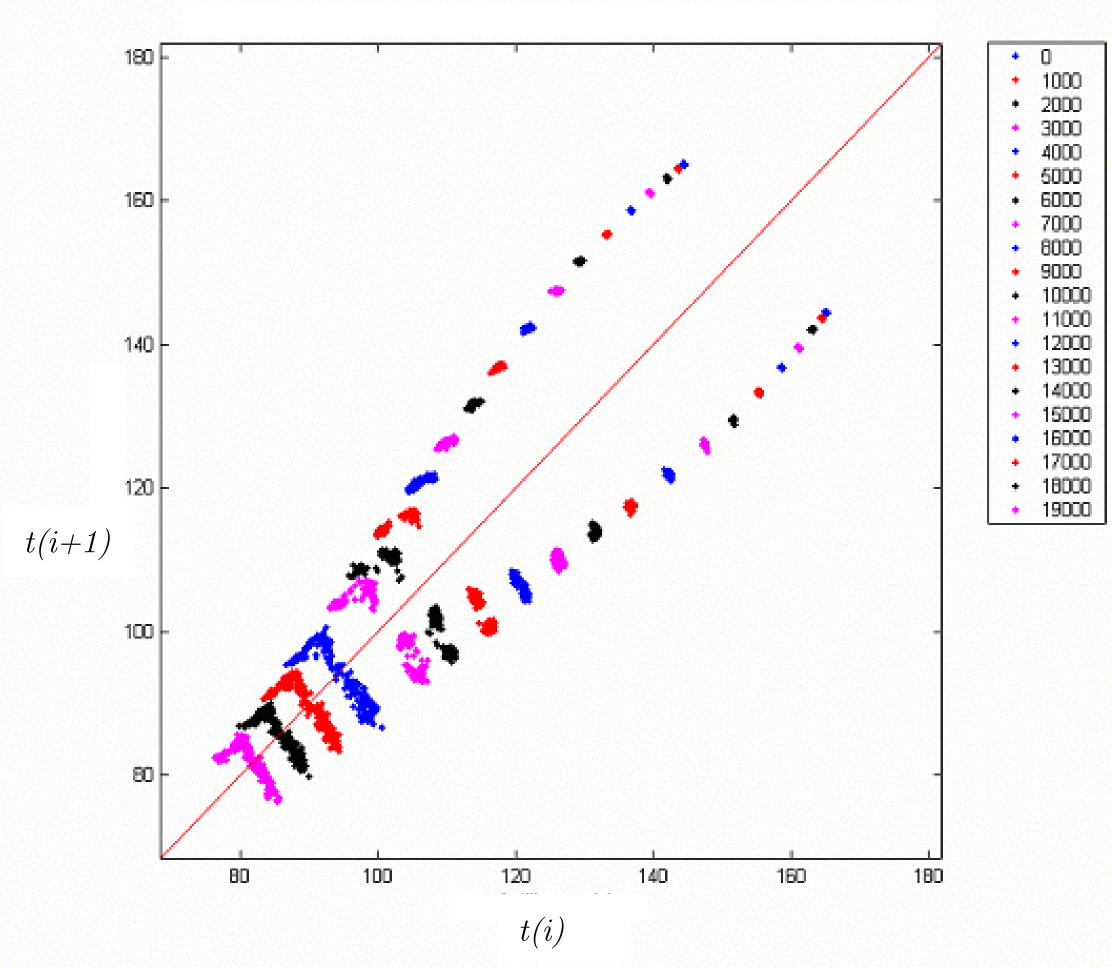


Figure 3.4: Return maps for a bubble column time series

The gas flow rate was 170 cc/min for all cases. The electrostatic potentials (shown in the legend at right) ranged from 0 through 19000 V. As the electrostatic potential was increased, the bubble rate increased or the time-interval between the successive bubbles decreased. However, note that the bubbling is faster but more complex since many more inter-bubble intervals are possible. The units for the ordinate and abscissa are milliseconds. A bifurcation diagram in this context is the projection of all these points to a 135 degree line.

3.2. The fluidized bed

A fluidized bed typically consists of a vertically oriented chamber, a *bed* of particulate solids, and a fluid flow distributor at the bottom of the chamber. The fluid flows upward through the particles, creating a drag force that counteracts gravity. With sufficiently high flow, the solids are levitated and move in complex, turbulent patterns (hence the name "fluidized"). This turbulence promotes heat and mass transfer as well as chemical reactions between the fluid and the solids. As the fluid flow rate is increased, the small amplitude highly complex behavior gives rise to large-amplitude approximately periodic behavior. With further increase in the gas flow rate, the approximately periodic behavior is interrupted by "stutters", and finally yields to turbulence.

This section and the fluidized bed data used in this study are taken from Daw et al (1995). The fluidized bed in Daw et al (1995) was a cylindrical vessel 10.2 cm in diameter, and the settled bed height was 23.5 cm. The particles used in the experiments described here are uniform 4.5 mm diameters steel spheres. Room temperature air was metered at constant flow into the plenum chamber below the gas distributor. Figure 3.5 shows a schematic of the fluidized bed setup.

The measurements made on bed dynamics were pressure differentials between flush, wall-mounted taps located 10 and 23 cm above the air distributor, respectively. Analog signal form the pressure transducers were bandpass filtered (0.1-40 Hz) to remove DC bias, prevent aliasing, and remove any contamination with 60 Hz noise associated with nearby AC equipment. The particles were classified as Geldart type D according to the fluidized bed literature. Twenty time series were collected for various gas

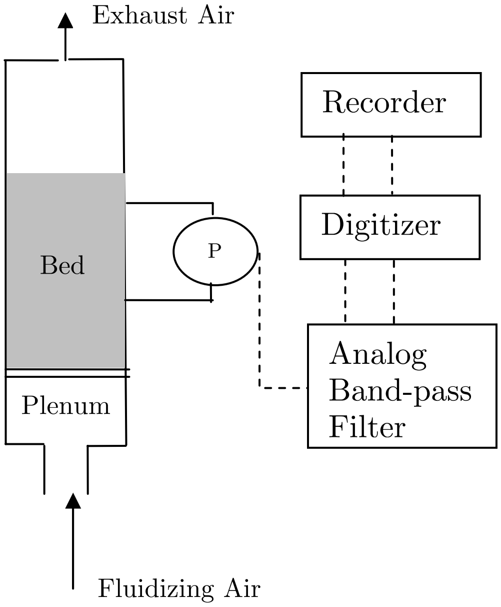


Figure 3.5. Experimental fluidized bed setup.

Redrawn from Daw et al (1995)

flow rates and the behavior captured in them ranged from approximate periodicity to turbulence.

Figure 3.6 shows four time series from the fluidized bed. All segments contain 2000 records and correspond to a time window of 20 seconds. Figure 3.7 shows the power spectral densities for the time series shown in figure 3.6. A 8192-point FFT was calculated (windowed with done with a symmetric 4096-point Hanning window). The spectral density was calculated for 6 non-overlapping time series segments, each 8192 records long. For the process, every segment represented the time window of 81.92 seconds. The solid lines depict the average power, and the dashed lines depict 95% confidence intervals for the power. Note that the 95% confidence intervals are very wide at the peaks, and that as the bed becomes more turbulent, the distinct peaks in the spectral power disappear.

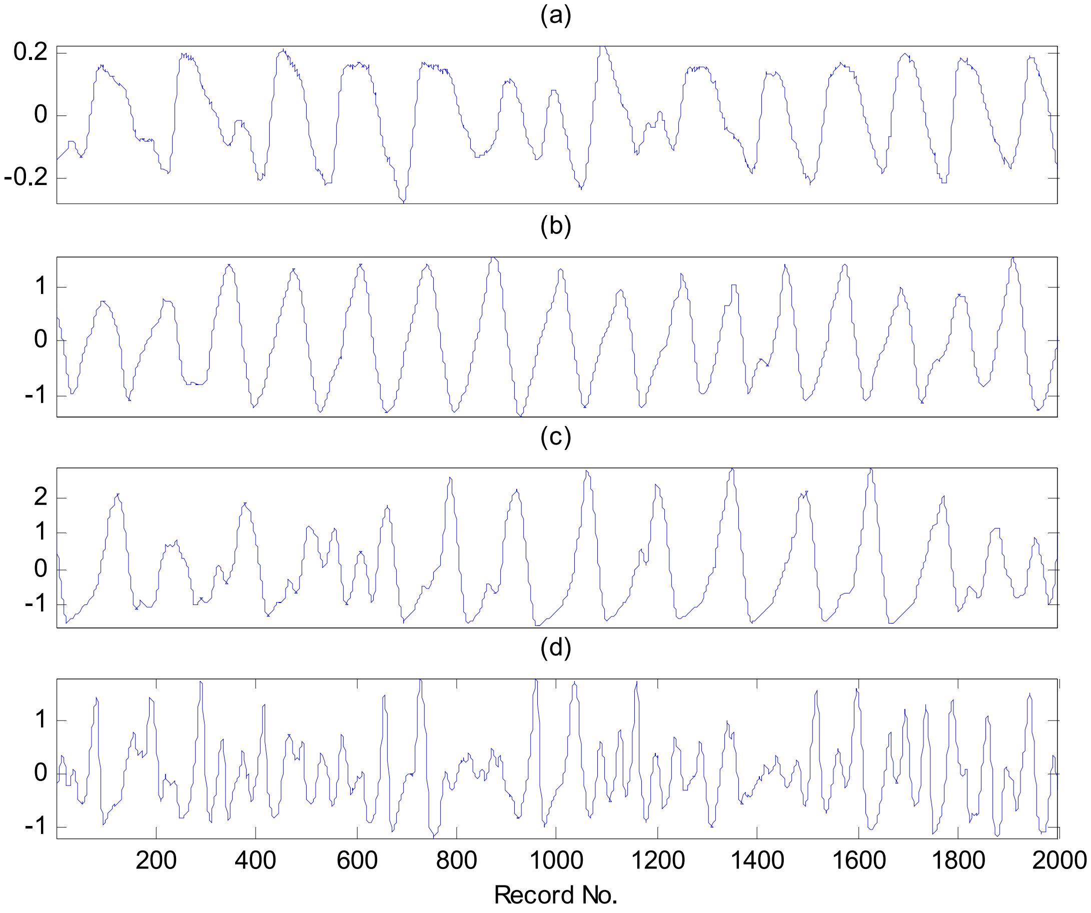


Figure 3.6. Differential pressure time series from the fluidized bed

From top to bottom (a) low-amplitude complex behavior (b) approximately periodic behavior (c) approximately periodic behavior interrupted by "stutters" (d) nearly turbulent conditions. Every subplot contains 2000 records and covers a time window of 20 seconds. The abscissa for every subplot is differential pressure. See text for details.

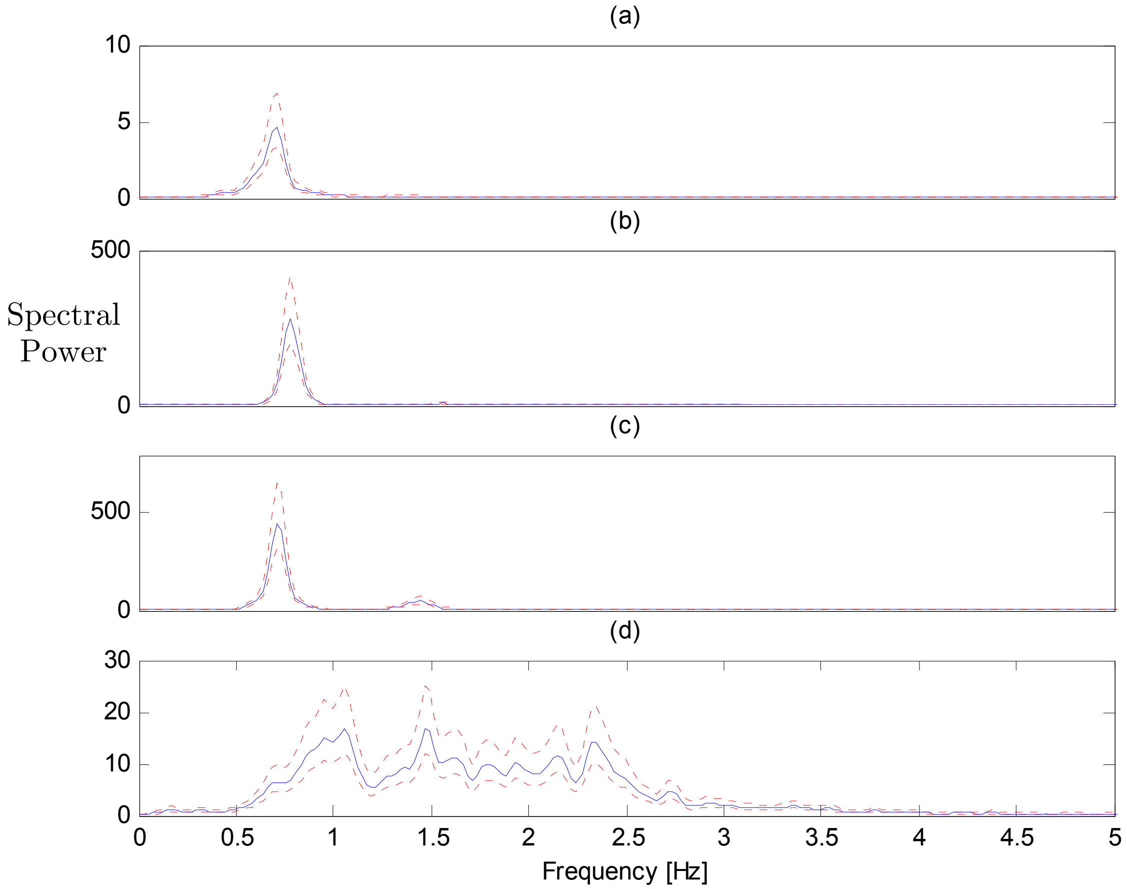


Figure 3.7. Spectral densities for fluidized bed time series.

The subplots (a) through (d) correspond to the time series shown in figure 3.6 (a) through (d). The abscissas for all subplots are the spectral power density for the windowed FFT referred to in the text. In all cases, the confidence limits were computed by considering six non-overlapping 8192-point time series segments, and assuming that the mean power at each frequency was distributed as Student's t-statistic.

Everything should be made as simple as possible, but not simpler.

-Albert Einstein

Chapter 4

Cluster-linked principal curves

4.1. Introduction

This chapter is discusses characterizing a time series by the probability distribution either of the measurements themselves or of the time-based return maps extracted from them. The concepts of cluster-linked principal curves and interpolating splines are introduced, discussed and developed. The next chapter exploits principal curves to compare the dynamics of two different time series and to test for stationarity and reversibility.

A distribution of data points can be defined in many ways. A linear gaussian random process (LGRP), for example, is defined completely by its mean and variance-covariance matrix. Another way to characterize a time series is by modeling it. In the usual statistical setting, the variables are neatly divided as independent and dependent variables. Linear regression techniques can then find a linear arrangement of the independent variables that explains the variation in the dependent variable. Linear regression can also be used when the functional relationship between the independent and dependent variables is nonlinear as long as the function has finite discontinuities. The term *linear* in that case signifies that the regression coefficients enter the regression equation linearly. The trick is to transform the independent or predictor variables into the pre-defined functions and to treat the transformed variables as *independent* variables. Obviously, this approach depends heavily on picking the appropriate transforms. If there is no strong evidence for a particular form or expression, or if many redundant transformations are introduced, the model converges poorly or not at all, and often is very unstable to be of any practical use.

Often one is faced with multivariate data sets and the dichotomy between dependent and independent variables is not apparent. The goal in that case is provide a summary of the data, while preserving most of the information present in the measurements. In that case, minimizing a *model error* is not an issue.

4.2. Principal component analysis

Principal Component Analysis (PCA) is probably the most well known and widely used multivariate statistical technique. The relationship between two different, random variables is quantified by cross-correlation. PCA exploits the cross-correlation to find the linear combinations of these variables, or directions that are associated with high variation. These directions are called the principal components (PCs), and are mutually orthonormal. Each of these directions has a corresponding eigenvalue that is a measure of variability along it. The eigenvectors or principal components are so ordered that the first eigenvector pertains to the largest eigenvalue, the second eigenvector to the second largest eigenvalue, and so on. This provides a hierarchical and orthonormal basis for the data space, with each Principal component explaining the maximum remaining variance.

If **C** is the variance-covariance or correlation coefficient matrix of the measurements, then it can be written as[5]

$$\mathbf{C}=\mathbf{U\Sigma U}' \tag{26}$$

where **U** contains the eigenvectors (or singular vectors) of **C**, and **Σ** is a diagonal matrix with the corresponding eigenvalues along the diagonal. ■

If $\mathbf{u}_1$ is the first column of **U**, or the first Principal Component (PC), then $p_1=\mathbf{x}'\mathbf{u}_1$ is the first Principal Score (PS), $p_2=\mathbf{x}'\mathbf{u}_2$ is the second PS, and so on.

PCA can be used for several ends. It reduces the dimensionality of the data and facilitates visualization. It can also be used to remove the noise by setting the principal scores corresponding to small eigenvalues as zero and projecting the principal scores back to the original basis of the data space. Note that **U** is an orthonormal matrix and $\mathbf{U}^T\mathbf{U}=\mathbf{I}$. However, if the relationship between the measurements constituting the multivariate vector is not linear, cross-correlation is not a suitable representation of the relationship between the variables, and PCA may not be appropriate.

There have been many extensions of PCA. An early example can be found in Gnanadesikan (1964) where generalized PCA is suggested. Gnanadesikan suggests transforming the variables such that the transformed vector contains cross-products and higher order polynomials of individual

[5] This decomposition is true only for symmetric matrices. Covariance matrices are symmetric by definition.

variables. The argument is that the resulting space contains only linear cross-correlations and PCA will be applicable.

Recently, there has been a great interest in local[6] PCA where the data space is divided in some zones, each of whom can then be summarized or reduced in dimension with a localized PCA. Many researchers have posed localized PCA as an optimization problem or a neural learning problem. Local PCA has been also posed as a mixture-model problem. However, local PCA depends on the partition of the data space, and though useful in solving complex pattern recognition problems, the end-product is not a smooth curve summary of the data.

Local PCA has also been studied in the context of nonlinear dynamics and chaos. Several researchers have suggested using localized PCA on the wavelet transforms to remove noise in chaotic time series. Kostelich and Yorke (1988) discuss localized PCA for smoothing trajectories in context of nonlinear dynamics. Locally weighted regression has also been an active area of statistical research. For a good review, see Atkeson, Moore and Schaal (1996).

However, most local PCA or local learning methods are useful but this dissertation is concerned with providing a smooth, continuous summary curve of the data, which is not achieved by these methods do not achieve. The following section discusses how to get a global summary of data as a polygonal line.

[6] Equation (25) is an example of global PCA where all data points are considered. Local PCA relies on using the data in some pockets and performing PCA on the smaller pockets.

4.3. Principal curves

A *Principal Curve* is a hyper-curve that locally approximates the data points [Hastie and Stuetzle (1989)]. The curve is data-driven and non-parametric; and can bend to the local density of the distribution. Henceforth Hastie and Stuetzle's Principal Curve is simply referred to as HSPC.

Figure 4.1 is a graphic depicting the central idea of principal curves. Least Squares Regression (LSR) attempts to minimize the 'error' in the predicted variable. PCA on the other hand minimizes the sum of squared (orthogonal) distances (SSD) of points from a straight line. Principal Curves, it can be seen in the figure, minimize the SSD of points from a *curve*.

In the definition of Hastie and Stuetzle, a HSPC is self-consistent, i.e., any point on the curve is the expected value of the distribution at that point. It is a generalization of PCA, but the straight line is replaced by a 'curve' that attempts to explain a large part of the variability present in the data. One would like to impose such conditions on such a curve. It should:

1. Pass through the center of the data cloud;
2. Be continuous;
3. Change considerably only in a region geometrically close to the region where some points are added or removed (local, not global);
4. Be determinable in a non-parametric way, i.e., not involving any restrictions about the distribution.

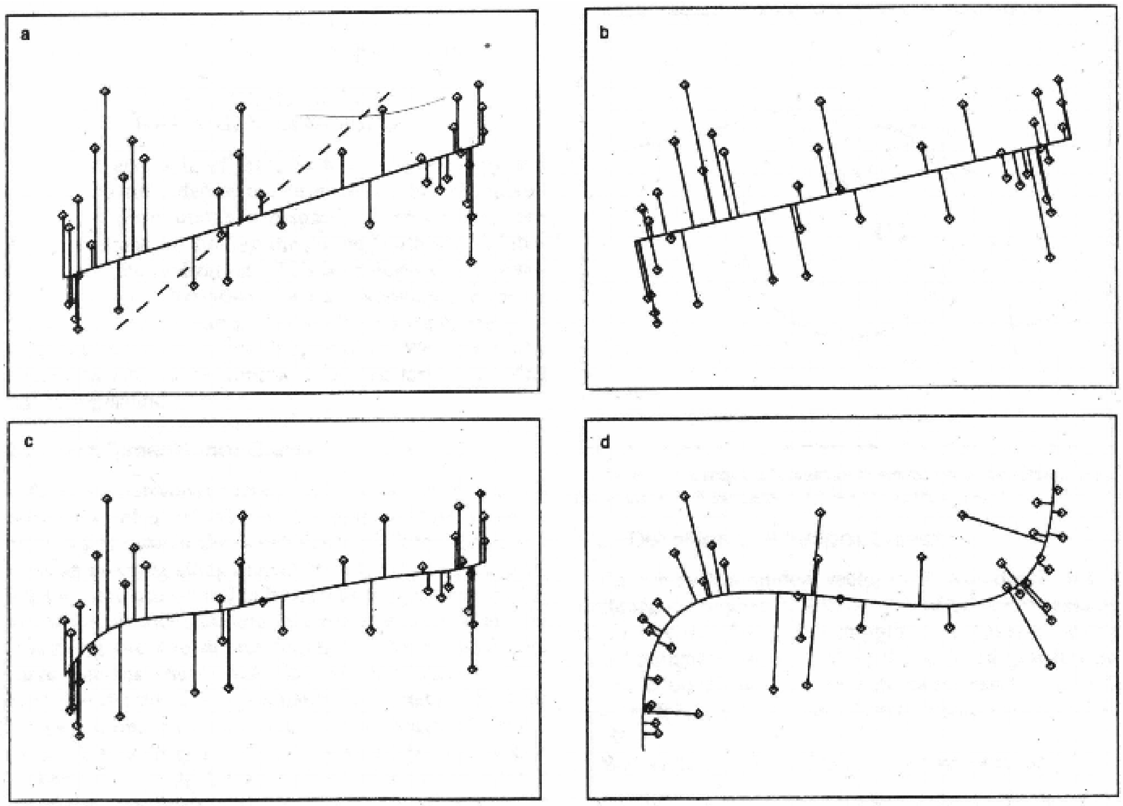


Figure 4.1: Graphic to demonstrate principal curves

(a)Regression line minimizes SSD in the dependent variable (ordinate) (b)PCA minimizes SSD in all variables (ordinate as well as abscissa) (c)A smooth regression curve minimizes SSD in the response variable, subject to smoothness conditions (d)The Principal Curve minimizes SSD in all variables, subject to smoothness constraints

Reproduced from Hastie and Stuetzle (1989)

4.3.1. History of principal curves

The original definition of Principal Curves by Hastie and Stuetzle is as follows.

> Denote by **x** a random vector in R^p with density *h* and finite second moments. Without any loss of generality assume E(**x**)=0. Let **f** denote a smooth (C^∞) curve in R^p parameterized over $\Lambda \subseteq R^1$, a closed (possibly infinite interval), that does not intersect itself $\lambda_1 \neq \lambda_2 \Rightarrow f(\lambda_1) \neq f(\lambda_2)$ and has finite length inside any finite ball in R^p. The projection index is defined as λ_f: $R^p \rightarrow R^1$ as
>
> $$\lambda_f(\mathrm{x}) = \sup_{\lambda}\{\lambda : \| x - f(\lambda) \| = \inf_{\mu} \| \mathrm{x} - f(\mu) \| \} \tag{27}$$
>
> The projection index λ_f(**x**) of **x** is the value of λ for which **f**(λ) is closest to **x**. If there are several such values, the largest one is picked. For proof about the existence and measurability of λ_f(**x**), see Hastie and Stuetzle (1989).■

The projection index is called the *arc length*. Hastie and Stuetzle essentially define their principal curve as a curve parameterized by λ, which is the length of the curve from its beginning point to the point on the curve where **x** projects –or the *arc length*. The question here is to reduce a multivariate[7] vector **x** to a certain λ value. The HSPC algorithm is as follows:

[7] Hastie and Stuetzle focus their attention on smoothing two dimensional scatterplots. Hastie and Tibshirani (1990) discuss a general class of models called Linear Additive Models. The application of HSPC algorithm to data with dimensions larger than two is discussed later in this chapter.

Initialization: Set the principal curve as the first PC of the distribution so that it passes through the center of the data.

Over iteration counter j, repeat

1.Set $\mathbf{f}^{(j)}(\mathbf{x}) = E(\mathbf{x} \mid \lambda_{f^{(j\text{-}1)}} = \mathbf{x})$

2.Define $\lambda^{(j)}(\mathbf{x}) = \lambda_{f^{(j\text{-}1)}}(\mathbf{x}) \forall \mathbf{x} \in h$

3.Evaluate $D^2(h, \mathbf{f}^{(j)}) = E_{\lambda_{f^{(j)}}} E(\| \mathbf{x} - f(\lambda^{(j)}(\mathbf{x})) \|^2 \mid \lambda^{(j)}(\mathbf{x}))$

Until $| D^2(h, \mathbf{f}^{(j)}) - D^2(h, \mathbf{f}^{(j-1)}) | \,/\, | D^2(h, \mathbf{f}^{(j-1)}) |$ is less than a prescribed threshold.■

The algorithm involves an Expectation-Maximization (E-M) procedure. After every iteration the arc lengths are reset so that the minimum arc length is zero. Hastie and Stuetzle do not provide a proof for existence or convergence of principal curves, but state that their implementation usually works.

The algorithm consists of two steps, namely those of projection and smoothing. In the projection step, all data points are projected on the principal curve and a corresponding arc length (line integral from the beginning of the curve to the point where a data point projects on it). The second step redefines the curve based on the arc lengths of data points. The data points are so arranged that their arc lengths are increasing. This step defines a polygon which is formed by connecting the points ordered by their arc length. The curve is then evaluated for self-consistency, by projecting the data points on the redefined curve. When the curve is self-consistent, the algorithm is assumed to have converged.

The existence of principal curves for non-trivial distributions has been studied by Duchamp and Stuetzle (1996A) who studied principal curves in a plane. They found the solutions to differential equations for uniform densities on rectangles and annuli, and discovered oscillating principal curves in addition to straight line and circular ones. Their work showed that principal curves are not unique. The HSPC algorithm converges to a local minimum of the distance function, and may or may not provide a meaningful solution in general. Duchamp and Stuetzle (1996B) showed that all principal curves are saddle points of the distance function –which is tantamount to their being a local minimum and not a global one.

Several papers published after the seminal paper of Hastie and Stuetzle approach the problem from different viewpoints, mainly to more rigorously gauge the existence, and convergence and bias of principal curves – the issues noted in the original paper by Hastie and Stuetzle. The approach taken by Tibshirani (1992) is semi-parametric, and involves maximizing the likelihood ratio based on the ref. Dempster, Laird and Rubin (1977). Kégl (2000) treats Principal Curves as an unsupervised learning scheme, and introduces principal curves of a fixed length. Delicado (2001) proposed another definition based on a property of the first principal components of multivariate normal distributions. He introduced the concept of Principal curves of oriented points (PCOP) where any point on the curve is the mean of the points in a hyperplane to which the curve is orthogonal or normal. Examples from these above-mentioned approaches perform roughly as well as the HSPC algorithm, but are computationally much more intensive. The approach of Hastie and Stuetzle is pursued in the further discussion, since theirs is the most basic and intuitive approach that yields satisfactory results. Besides, the other approaches have not been shown to yield superior results in comparison to HSPC algorithm.

The most serious practical issue with the HSPC algorithm (and all other algorithms proposed for principal curves) is that the conditional expectation is not defined well for very low probabilities. In step 2 of HSPC algorithm, a point on the curve corresponding to an arc length of λ is the average of all other points that have the same arc length. In most cases, there is only one point (and most often none) at an arc length. To deal with this, Hastie and Stuetzle suggest using locally weighted running lines smoother as described in Cleveland (1979) or cubic smoothing splines according to Silverman (1985). Smoothing splines are defined as connected piecewise polynomials that satisfy some smoothness conditions and minimize the cost function of the form:

$$G(\mathrm{f})=\frac{1}{n}\sum_{i=1}^{n}\left|\mathrm{x_i}\text{-}\mathrm{f}(\lambda_i)\right|^2+\mu\sum_{i=1}^{n}\left|\mathrm{f}''(\lambda_i)\right|^2 \tag{28}$$

The cost function essays to reach a compromise between fit and smoothness. It is very difficult to use splines for a distribution with dimension greater than two. For that reason, smoothing splines are not considered the general discussion of principal curves.

The former method [Cleveland (1979)] is similar to iterated weighted least squares (IWLS), and using it in the HSPC algorithm replaces a point on the principal curve by the IWLS estimate for a cluster of points in its *neighborhood*. One has to decide upon a parameter, called *span* that decides how close two points are. The principal curve is a n_c-tuple $\{(\lambda_1, \mathrm{f}(\lambda_1)),(\lambda_2, \mathrm{f}(\lambda_2),\ldots,(\lambda_{nc}, \mathrm{f}(\lambda_{nc}))\}$ connected by straight lines. The curve can alternatively be written as an assortment of line segments $\{s_1, s_2,\ldots, s_{nc\text{-}1}\}$. HSPC is essentially a polygonal line, of which each vertex is practically a weighted cluster center.

4.4. Cluster-linked principal curves

Obviously, in the HSPC algorithm one has to specify n_c or the number of cluster centers to define the polygonal line. HSPC algorithm computes the

IWLS estimate for *every point*, thus rendering n_c=n. The span has to be specified also for defining the weight function or kernel.

In absence of extreme outliers, the IWLS estimate will be quite close to the cluster center, and the cluster mean can be used as a convenient and reasonably accurate substitute for the points on the principal curve. This obviates the need for IWLS and reduces computational cost. For n data points, the complexity of kernel-type smoother is O(n^2), which reduces to O(n) when the mean is used. The complexity of projection also decreases to O(n*n_c) from O(n^2) which is the case for HSPC algorithm.

Using the mean of the cluster offers huge reduction in computational cost. Hastie and Stuetzle note that their algorithm has a bias with respect to the true principal curve, but that the bias reduces as the density of data points increases. The data (real or experimental) encountered in practice contain noise though, and a small bias shouldn't hamper the success of the principal curve in describing the data. By bias a local bias is meant and not a global one. If the principal curve has a global bias compared to the distribution of data points, the curve is not self-consistent. Using fewer vertices may not be able to exactly reproduce the local gradient, but the bias would be local, and perhaps will cancel out. Later the issue of bias and mean of residuals is discussed in more detail.

It is proposed to have considerably fewer vertices or cluster centers in the polygonal line principal curve than the number of data points. The resulting curve is called *Cluster-Linked Principal Curve* (CLPC) since it essentially involves formation of clusters based the principal curve parameterization and redefining principal curves based on these clusters.

There are significant differences between CLPC algorithm and the HSPC algorithm. The reduction in computational cost has already been discussed. The salient difference is that HSPC algorithm smoothes each dimension separately, which is not exactly in keeping with the intuitive appeal of the Expectation-Maximization principle. For example, a data point may have a different corresponding arc length in X-Y plane than it would in say X-Z plane. This makes it impossible to define an arc length for every data point. Moreover, it is not desirable to make the choice of dependent and independent variables that one cannot avoid when smoothing has to be performed. Smoothing every dimension separately involves fitting a principal curve to two dimensions, and ignores the additional information present in other dimensions. The CLPC algorithm has only one hyper parameter, which is the number of vertices in the polygonal line, whereas HSPC algorithm required adjusting the spans of the kernel smoother (for all the dimensions).

One could argue that CLPC may suffer from imprecision in approximating the density of data points. However, Hastie and Stuetzle suggest using a span large enough to cover 70% of the data range at first, which has much more of a smoothing effect than that obtainable by localized clustering based on the arc lengths. On the other hand, using a large span oversmooths the scatterplot and may even remove finer structure. The accuracy of the principal curve can be enhanced by either increasing the number of clusters or by having more data. The algorithm is now outlined.

4.4.1. CLPC algorithm

Initialization: Set the principal curve as the first PC of the distribution so that it passes through the center of the data.

If the first principal component is **u**, then compute the arc lengths of points as

$\lambda_f^{(1)}(\mathbf{x}(i))$=$\mathbf{x}(i)$'*$\mathbf{u}$

Over iteration counter j, repeat

1.Expectation Step

Sort the data points so that $\lambda^{(j)}(\mathbf{x}(1))<\lambda^{(j)}(\mathbf{x}(2))<...<\lambda^{(j)}(\mathbf{x}(n))$

for k=1 to n_c,define the n_c cluster centers as:

$\mathbf{f}^{(j)}(k)=\mathrm{E}(\mathbf{x}(i)|\frac{(k\text{ - }1)n}{n_c}+1\leq i<\frac{kn}{n_c})$

Set the minimum arc length to zero

Define the arc lengths of the first cluster center as

$\lambda_f^{(j)}(1)=\mathrm{E}(\lambda^{(j)}(\mathbf{x}(i))\mid 1\leq i<\frac{n}{n_c})$

Define the arc lengths of other cluster centers as

$\lambda_f^{(j)}(k)=|\lambda_f^{(j)}(k)\text{-}\lambda_f^{(j)}(k\text{ - }1)|+\lambda_f^{(j)}(k\text{ - }1)$

Define line segments as $\mathbf{s}_k^{(j)}=[\mathbf{f}^{(j)}(k\text{ - }1),\mathbf{f}^{(j)}(k)]$

2. Projection Step

Find the arc length and its orthogonal distance from the nearest line segment for all points (See appendix A)

Compute:

$\lambda^{(j)}(\mathbf{x}(i))$, the arc length of $\mathbf{x}(i)$

$d_f^{(j)}(\mathbf{x}(i))$, its orthogonal distance from the nearest line segment

3.Evaluate $\mathrm{D}^2(\mathrm{h},\mathrm{f}^{(j)})=\sum_{i=1}^{n}\left(d_f^{(j)}(\mathbf{x}(i))\right)^2$

Until $\frac{|\,D^2(\mathrm{h},\mathbf{f}^{(j)})\text{ - }D^2(\mathrm{h},\mathbf{f}^{(j\text{-}1)})\,|}{|\,D^2(\mathrm{h},\mathbf{f}^{(j\text{-}1)})\,|}$ is less than a prescribed threshold■

The principal curves are not well defined near the extremities, and the reason is the familiar issue of extrapolation to which no satisfactory answer can be given. In our implementation, the vertices at the extremities of the CLPC are formed by assigning them only half as many points as other cluster centers. Another way to deal with this issue is to loop the principal curve by closing the polygonal line by joining the last cluster center to the first. A principal curve defined that way will be able to approximate even closed structures. However, one should not try to fit closed polygonal line CLPC if the data are approximately monotonic as it may lead to convergence problems for obvious reasons.

Another important point is the orthogonal projection on the line segments. Although not very likely, it is possible that a data point is not orthogonal to any of the line segments. It is also possible that a point is orthogonal to a line segment but outside its endpoints. Figure 4.2 shows an example of the latter.

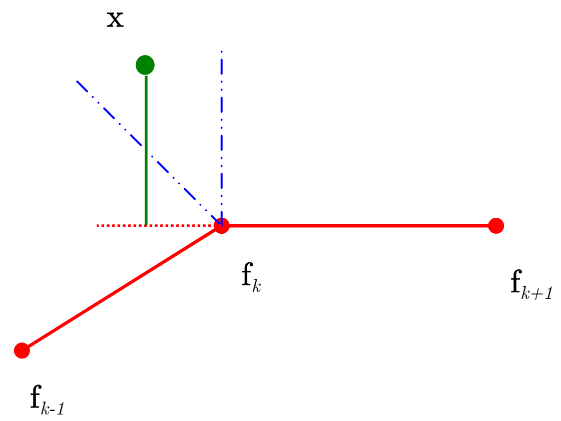


Figure 4.2: Projecting a point on the principal curve.

For a detailed treatment, see appendix A.

Any point that lies in the zone between the two dash-dot lines will not be orthogonal to the line segments in such a way that its projection lies within the endpoints of that line segment. There are two ways to deal with it. The point can be projected to the closest vertex of the polygonal line and assigned an arc length corresponding to that vertex. The more jagged the polygonal line, the more problems will arise due to this approach. The second way is to accept the new arc length obtained by extrapolating the line segment to which the point is closest, and thereby accept a small amount of error in the estimation of arc lengths. The second approach is followed in our implementation. By increasing the number of cluster centers the curve can be made smoother and the likelihood of the possibility delineated in figure 4.2 reduced.

If the first principal curve does not explain enough variation in the data set, another principal curve can be fitted to the residuals. In our implementation, variability is approximated by *generalized variance*, or the sum of the eigenvalues of the covariance matrix of the residuals or that of the data. Section 4.4.2 concerns itself with residual analysis.

Figure 4.3 shows how principal curves can approximate a noisy parabola. The parabola was defined as $y=4x(1-x)+e_i$ for $0\leq x\leq 1$ where $e_i \approx N(0,0.15^2)$ is random Gaussian noise with mean of zero and standard deviation of 0.15. The data were then scaled to have zero mean and unit variance, or scaled to Z-scores. The arc begins at the bottom left of the figure and ends at bottom right. 17 cluster centers were used for the approximation. The fit doesn't seem to have a noticeable bias and is quite smooth. It also passes through the center of the data cloud.

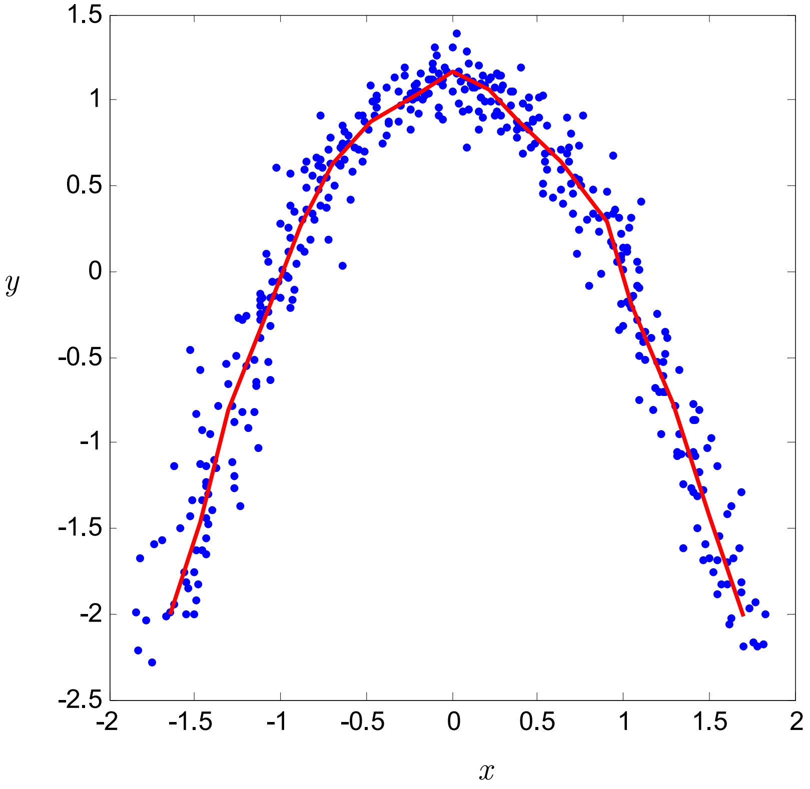


Figure 4.3. Principal curve for a noisy parabola

Figure 4.4 shows the residuals obtained after fitting the principal curve. The residuals appear to be randomly distributed. Most of the residuals are contained within a square with a side of 0.3 whereas the data fitted rested within a square with a side of 4 units. The area occupied by the residuals is therefore around more than 150 times smaller than that occupied by the original data. Based on the fit seen in figure 4.3 and the residuals seen in figure 4.4, it seems that the principal curve described the variability in the data set quite well and the residuals are white noise. Now the issue of residual analysis is taken up.

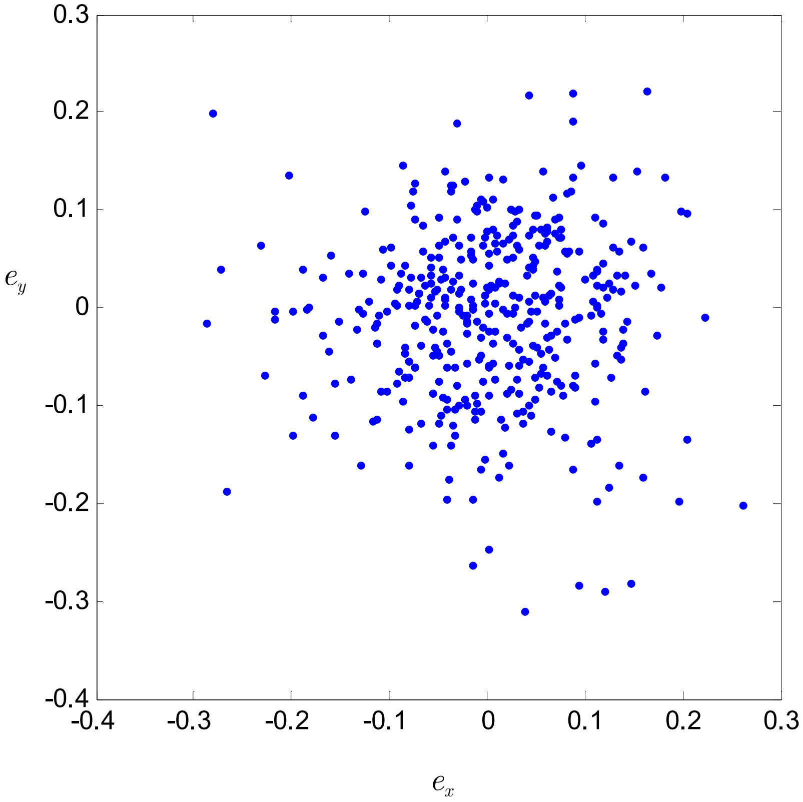


Figure 4.4. Residuals of the fit in figure 4.3

4.4.2. Analysis of residuals

After arriving at the residuals, there are three things that are desirable and that need to be examined. They are as follows:

1. The mean of the residuals should be zero, or nearly zero
2. The variability remaining in the residuals should be much smaller than that contained in the data set
3. The residuals should be independent of each other

These issues are now addressed. Note that no confirmatory analysis is performed, but some measures are suggested that quantify the departure from the desired conditions developed. First let us introduce some terminology.

Let $\hat{x}_i$ be the projection of a point $\mathbf{x}_i$ on the principal curve. In mathematical notation, $\hat{x}_i = E(x|\lambda(x)=\lambda(\hat{x}_i))$. The previous expression means that $\hat{\mathbf{x}}_i$ is the mean of all the points having the same arc length as itself. Let the residuals be defined as $\mathbf{e}_i = x_i - \hat{x}_i$, and the covariance matrix of residuals as $E = \mathrm{Cov}(\mathbf{e}_i)$, and the mean of residuals as $\boldsymbol{\mu}_e = E(\mathbf{e}_i)$. At the same time assume that the covariance matrix of data is $C = \mathrm{Cov}(\mathbf{x}_i)$ and the mean of the data is $\boldsymbol{\mu} = E(\mathbf{x}_i)$.

4.4.2.1. Zero mean

The fact of the mean being zero can be checked by Hotelling's T^2 statistic. The relevant statistic to be computed is $T^2 = \boldsymbol{\mu}_e^T C^{-1} \boldsymbol{\mu}_e$ and it is distribution is related to the F-statistic. This computed value is also known as Mahalanobis distance. A test is not encouraged but it is suggested to just look at the mean of the residuals and the Mahalanobis distance. In almost all our simulations the mean was nearly zero, and the issue of bias in the residuals is not deemed crucial for the CLPC algorithm. To conduct a test, the interested reader is referred to any standard text on multivariate analysis.

4.4.2.2. Remaining variability

A good estimate of the variability remaining in the residuals is the sum of eigenvalues of E, or the trace of E. It is desirable that the ratio of traces of E and C or trace(E)/trace(C) be small. The ratio also indicates the fraction of variability remaining in the residuals or the variability not explained by the principal curve. This concept is very similar to that of generalized variance and is used frequently in multivariate analysis.

4.4.2.3. Independence of residuals

Ideally, if the residuals are completely independent, **E** is a diagonal matrix. A measure can be suggested to ascertain that. The determinant of a diagonal matrix is equal to the product of its diagonal elements (or eigenvalues for that matter). The closer the determinant is to the product of its diagonal elements, the higher is the likelihood of the residuals being independent. Of course, it is not stated as a rigorous fact since it is possible that the determinant of a non-diagonal matrix is equal to the product of its diagonal elements. It is indeed possible, though not very likely, and it is only suggested that this measure be used in tandem with the previous two measures (Mahalanobis distance and fraction of generalized variance remaining). There are many tests in statistical literature to test for sphericity and diagonality, to which the interested reader may refer. The measure $\det(\mathrm{E})/\prod_{j=1}^{i=n}\mathrm{E}_{ii}$, which is the ratio of the determinant of **E** to the product of its diagonal elements, can be computed. The closer is this ratio to one, the more unrelated the residuals are.

Now let us compute the three measures defined above for the residuals from the fit in figure 4.3. The residuals themselves are plotted in figure 4.4. Mean of residuals ($\boldsymbol{\mu}_{\mathbf{e}}$) is $[0.0042\ 0.0038]^{T}$, the Mahalanobis distance ($\boldsymbol{\mu}_{\mathbf{e}}^{\mathbf{T}}\mathbf{C}^{-1}\boldsymbol{\mu}_{\mathbf{e}}$) is 0.0000321, and the remaining variability, which is (trace(**E**)/trace(**C**)), is 0.0782. Thus the mean of the residuals is almost zero, the Mahalanobis distance is very small, and only 7.82% of generalized variance remains in the residuals. The ratio of the determinant of **E** to the product of its diagonal elements $\det(\mathbf{E}) / \prod_{j=1}^{i=n} E_{ii}$ is 0.9988, which is very close to one. The off-diagonal elements of **E** are 80 times smaller than those on the diagonal, and the matrix is quite diagonal. Thus it appears that the residuals do not contain any pattern

4.4.3. Choosing the number of clusters

It was observed above that using a larger number of cluster centers reduces the bias in the fitted principal curve. Figure 4.5 shows the fitted principal curves using 7, 13, and 27 cluster centers. It is clear that using too few cluster centers impairs the ability of the algorithm to bend to the density of the data. On the other hand, using too many cluster centers results in the curve attempting to fit even the noise by attempting to visit each point very closely. The goal is to find the number of cluster centers so that the curve explains a large fraction of the variability in the data, but is also smooth. The number of cluster centers can be set so that the criterion described in equation (28) is minimized. Alternatively, informational complexity measures such as Akaike's Information Criterion (AIC) can be used to compare various principal curves obtained. AIC is made of two penalty terms that penalize badness of fit and excess parameters. The curve having the minimum corresponding AIC should be chosen as the most parsimonious and efficient model. Schemes like cross-validation can be used, but they are computationally very expensive. Another disadvantage to using cross-validation is that it is not very effective if there is not enough data. A CLPC is not a global minimum of the distance function, but a saddle point instead, which makes it difficult to compare the bootstrap estimates. However, the mean squared error for the training data and the test data can be compared to see if the curve is robust. There is however, no statistical test to validate

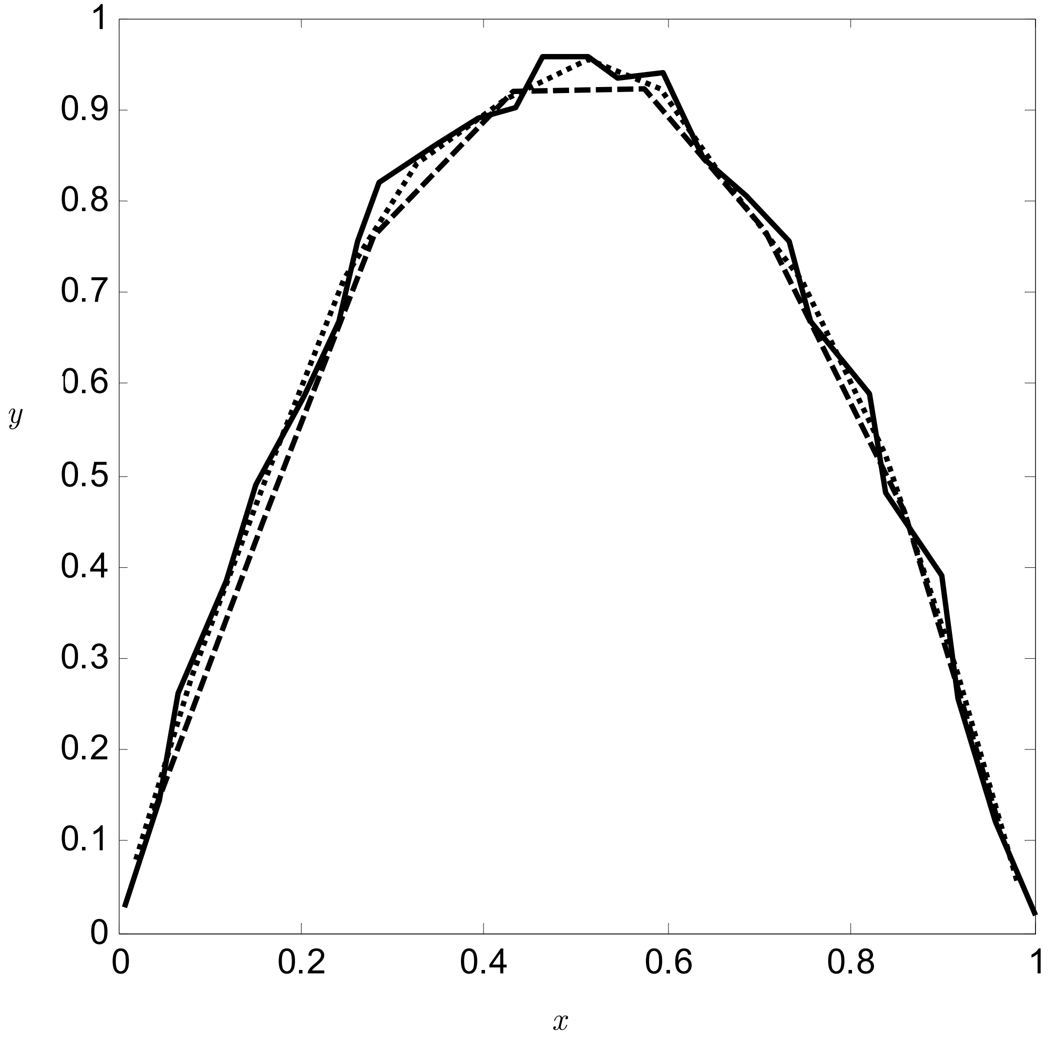


Figure 4.5. Fitted principal curves with various clusters

n_c=7 (dashed line), n_c=13 (dotted line) and n_c=27 (solid line) the model by comparing mean squared errors of training and test data unless one invokes the F-distribution. Variable selection methods can be invoked but they also rely on a partial F-test and are parametric whereas the CLPC is a non-parametric curve. Another way is to create a sort of SCREE[8] plot, and choose the number of clusters based on the 'knee' in the SCREE plot.

However, it must be stated that although it is desirable to know the optimum number of cluster centers, it is not essential. The aim of a CLPC is to provide a summary of data, and even if the number of clusters used is 20% more than optimal, it is not a major problem. Admittedly, the training time increases as the number of clusters is increased and the projection step becomes costlier. However the addition in cost is linear, and can be tolerated well. Nonetheless, using too many clusters may cause convergence problems since the curve may attempt to learn peculiarities of the data set in question. To explore all possible principal curves of a distribution is still an active area of research, and is beyond the scope of this study.

The best way to decide upon the optimum number of cluster centers is by visualizing the shape of the curve, since the human eye is adept at making trade-offs between smoothness and accuracy. However, that luxury is not possible if the dimension of the data is more than three. In that case, separate scatterplots can be used. Our experience indicates that the optimum number of cluster centers depends on the density of data points and the shape of the distribution. Using one cluster center for 10 to 20 points usually yields good results.

[8] A SCREE plot in this context is a plot of percentage of variability remaining versus the number of clusters. SCREE plots are used in Principal Component Analysis (PCA) to obtain the optimum number of Principal Components to be retained –which is taken to be the number that corresponds to the 'knee' of the plot, i. e., a point where the slope changes to a much smaller value. The principal components explaining little variation (to the right of the knee) are considered small, and not very important, like scree and not boulders.

Figure 4.6 shows how a closed or looped principal curve can approximate a noisy circle. The circle was formed by defining $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} sin(t) \\ cos(t) \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$ where $0 \le t \le 2\pi$ and $e_1, e_2 \sim N(0, 0.15^2)$. It appears that there is no significant bias in the fit. The residuals from the fit in figure 4.6 are plotted in figure 4.7.

Once again, the residuals appear to be randomly distributed. The area occupied by the residuals is 40 times smaller than that occupied by the data set (contained in squares of size 2 and 0.3 respectively). The mean of the residuals was $[0.0020 \quad -0.0018]^{T}$, the Mahalanobis distance was of the order of 10^{-5}, the fraction of generalized variance remaining was 0.79% and the ratio of determinant of **E** to the product of its diagonal elements was 0.999. The diagonal elements of **E** were roughly eighty times larger than the off-diagonal ones. Based on these measures, it appears that the residuals have no pattern and little variability compared to the original data set.

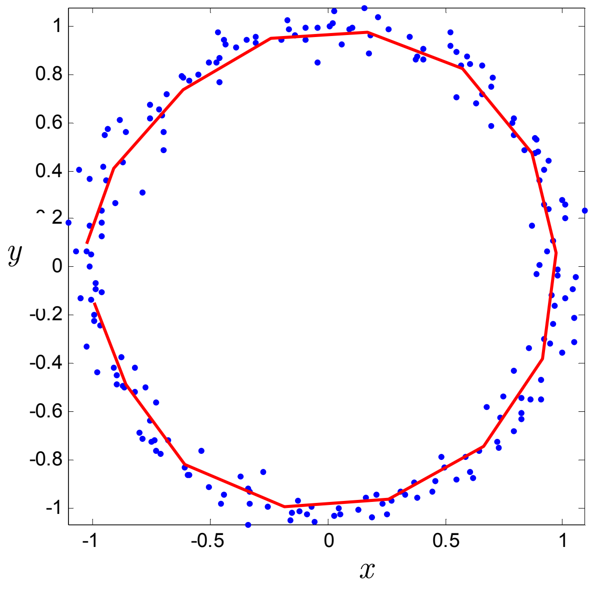


Figure 4.6. Principal curve fit to a noisy circle

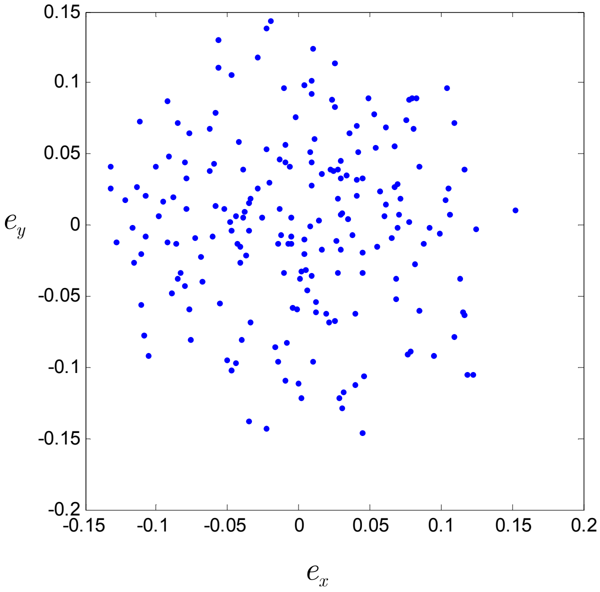


Figure 4.7. Residuals of the fit in figure 4.6

4.5. Characterizing an attractor from return maps

We will now show how principal curves can approximate the return map obtained from a bubble column[9] time series. Figure 4.8 shows the return map for the bubble column operating under chaotic conditions (see figure 3.3(d) for the time series). A principal curve was fitted to this data with 15 cluster centers. The result is shown in figure 4.9. It is apparent that the curve describes the distribution quite well. The residuals (not shown) do not exhibit any particular pattern, and it seems that the fit is good based on the three measures defined in section 4.4.2. Note that the inter-bubble interval *t(i+1)* is plotted versus *t(i)* in figures 4.8 and 4.9.

[9] The bubble column is described in section 3.1 of this dissertation

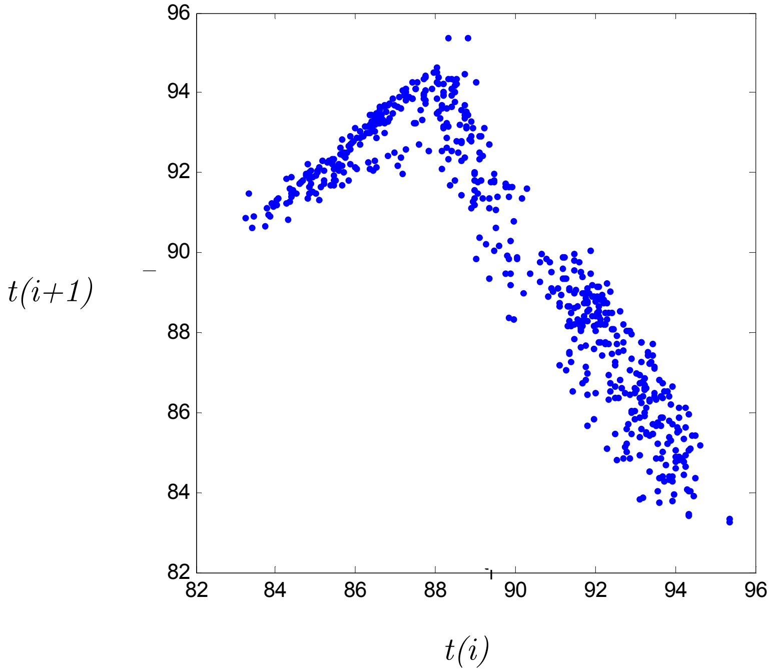


Figure 4.8. Return map for a bubble column time series

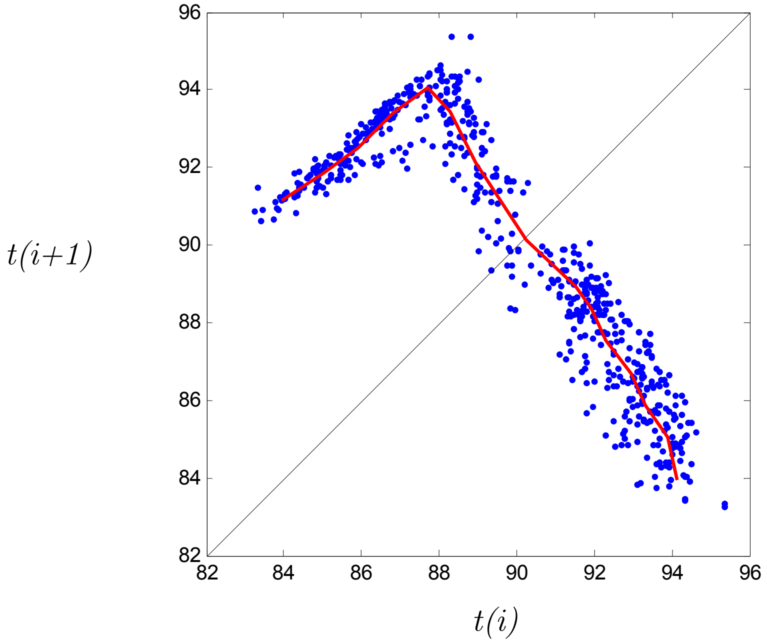


Figure 4.9. Principal curve fitted to the data in figure 4.8

To estimate the inter-bubble interval, the difference between successive upward crossings was used to obtain a more robust estimate. The units for ordinate and abscissa in figures 4.8 and 4.9 are milliseconds.

4.5.1. Interpolating splines and principal curves

In the special case of two-dimensional distributions, the principal curve can be smoothed additionally by way of splines. The CLPC is formed by connecting the cluster centers with straight lines. Using polynomials instead of straight lines can enhance the interpolation.

Splines are piecewise polynomials that have to fulfill certain conditions –namely those of continuity and differentiability. In this study third-degree polynomials are used, but the methodology can be extended to higher order polynomials. One has to be careful when using high degree polynomials owing to their sensitivity to a small change in the points they are supposed to fit. Information about fitting splines can be found in appendix B. Figure 4.10 shows the results of fitting an interpolating spline to the principal curve obtained on the return map from a bubble column time series (cf. figure 4.9).

A great advantage with using splines is that the situation outlined in figure 4.2 becomes much less likely since splines are smooth and continuous and their slope at the knots is continuous unlike that of a polygonal line at the cluster centers. However, using splines limits the algorithm to only two dimensions, which is a handicap. Splines can be used for each dimension separately but that approach is not in keeping with the concept of principal curves as passing through the center of the data cloud. The reason is that it is quite possible to have a point that is orthogonal to the splines fitted to X-Y and X-Z planes, at *different values* on the X-axis. For such a point, it is impossible to define the corresponding arc length.

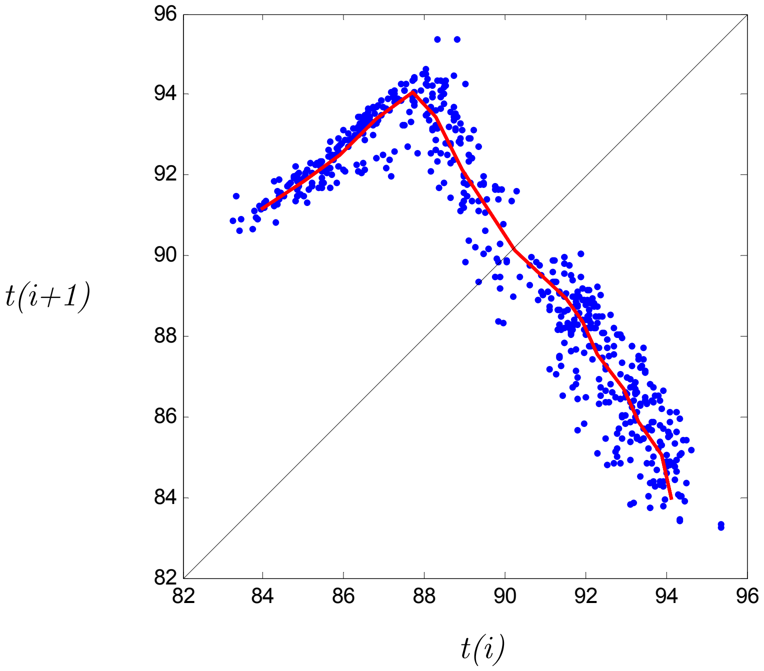


Figure 4.10. Interpolating spline for the fitted principal curve.

The units for ordinate and abscissa are milliseconds.

Spline surfaces can be used but they are computationally very intensive and do not reduce the dimensionality of the data, which is the primary concern of this study.

Many lower-dimensional chaotic systems can be adequately described in 2-D or 3-D return maps. Employing an interpolating spline after fitting a CLPC to the return map can help identify the fixed point of the systems as well as its stable and unstable manifolds.

Let us treat the spline in figure 4.10 as a reference. It is known that the fixed point lies on the diagonal. It is also known that the absolute slope of the spline at the point where it crosses the diagonal must be greater than one, since the system is chaotic. Thanks to the spline segment near the diagonal, a simple mapping of the form $x(i+1)=\mathrm{f}(x(i))$ develops. This map can be solved algebraically (third-order polynomials are being used here) or numerically. For example, the spline segment that approximates the region near the fixed point, is

$$t'(i+1)=-0.0116t'(i)^3+0.3688t'(i)^2-1.9822t'(i)+3.1732 \qquad (28)$$

where $t'(i)=t(i)-88.939$ and $88.939\leq t(i) \leq 90.2314$

Equation (28) allows one to find the fixed point and the approximate mapping near it, which can be profitably used to implement OGY or similar control schemes based on the return maps.

It is obvious that the coefficient of the cubic term is very small and can be ignored, whereupon the mapping reduces to a quadratic function of a form not very dissimilar to the logistic map. Fitting a cluster-linked principal curve and later an interpolating spline is not very time-consuming and can be done online. That allows one to apply adaptive control schemes as well.

The spline fitted to the principal curve can be used to iteratively generate a return map. Figure 4.11 shows the return map generated by iterating the interpolating spline for the fitted cluster-linked principal curve. The data in the figure is the same as that in figure 4.1, and corresponds to the mean crossings of a bubble column time series that exhibits a period-4 or a noisy period-4 behavior

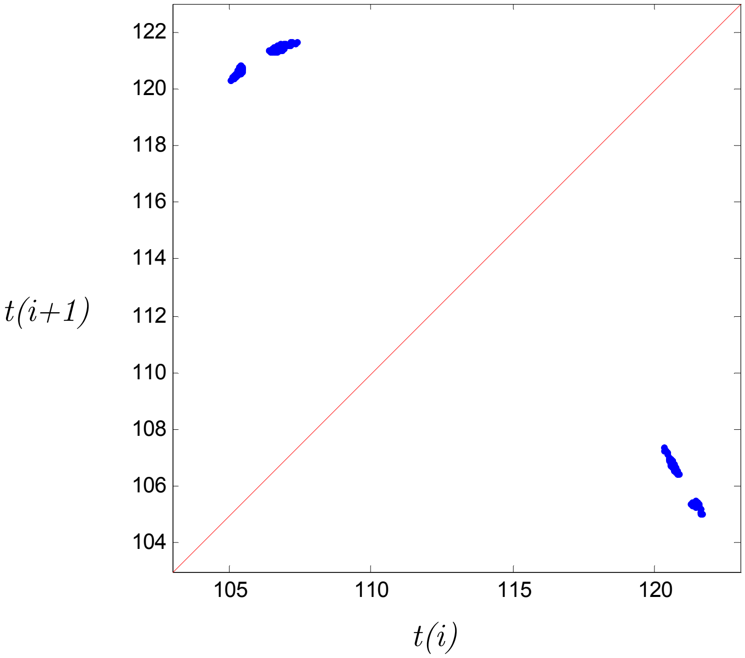


Figure 4.11. Return map obtained by iterating a spline.

The spline is shown in figure 4.10.

Figure 4.12 shows an overlaid return map where the blue dots are the data points in figure 4.11 and the red dots are the data points for the actual time series. It is clear that the return map obtained by iterating the spline preserves the general period-4 structure very well, and the iterated return map lies in the center of pockets of blue dots. Note that 2% noise[10] was added in generating figures 4.11 and 4.12. The effects of adding noise to the iterative map are explored below.

Figure 4.13 shows the results of iterating a spline fitted to the CLPC cluster centers (cf. figure 4.10). The series is studied in more detail in section 5.1 and figure 5.3. The red dots are the data points obtained by iterating the

[10] By 2% noise we mean that the standard deviation of the random noise added to the data was 2% of that of the data. The noise was added in the form $t(i+1)$=f($t(i)$)+e_i where e_i~N(0,(0.02σ)^2) where σ is the standard deviation of the measurements {$t(i)$}

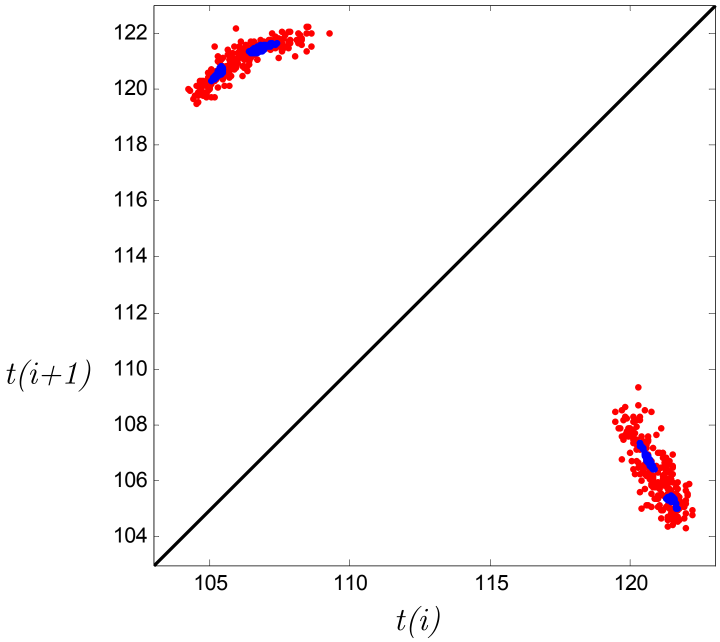


Figure 4.12. Overlaid return maps for a period-4 time series

The spline is shown in figure 4.11.

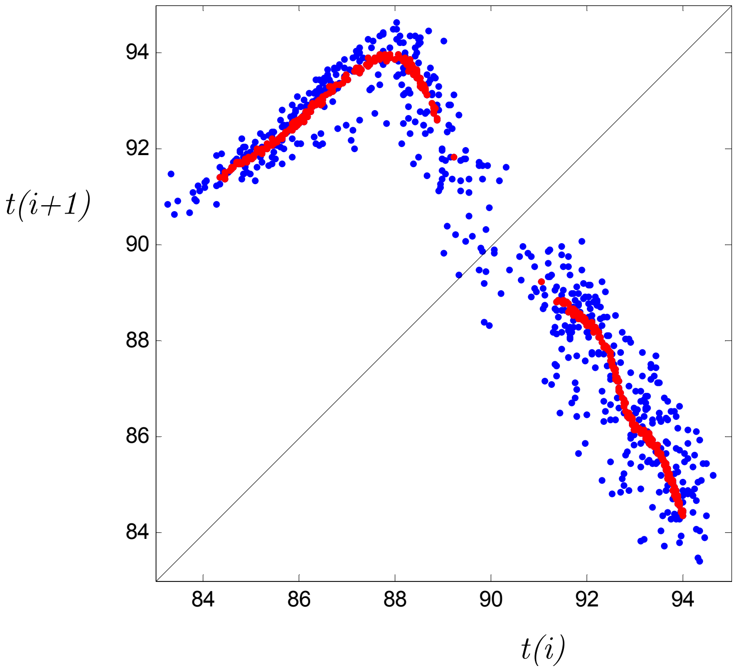


Figure 4.13. Return map for a chaotic time series with fitted spline

spline (no noise added in this case), and the blue dots denote the return map of the series. The return map obtained by iterating the spline, it can be seen, lies in the center of the true return map.

4.5.2. Effect of noise on spline-based return maps

The spline-based return map is simply a collection of connected polynomials, and all iterations fall on that piecewise polynomial or spline. Adding some noise to the iteration step may result in more realistic-looking return maps that approximate the scatter of the points in addition to describing their general shape. Figure 4.14 explores how adding noise to the iterative step can change the shape of the return map thus obtained. The return maps contain 400 iterations each. Note that the return maps seen in figure 4.14 do not depend on the starting point, if enough iterations are used. Even when the starting point was placed on the diagonal, the iterations spiral out of the zone near the diagonal and settle in the bands seen in figure 4.14(a).

Clearly the scatter of the data points increases considerably as more noise is added to the iterative step. Figures 4.14(b) and (c) approach the general pattern of points on the return map in figure 4.9. Adding too much noise however may destabilize the mapping because splines are not very accurate for extrapolation. Hints of the mapping becoming unstable can be seen in figure 4.14(d) where quite a few points fall far from the general spread of the points. Further research on fitting splines to a return map is underway.

As noted before, the approximation achieved by CLPC at any point is dependent on the local density at that point. In the above example, there are not a lot of data points near the diagonal line, and the approximation may not be accurate. It is up to the researcher to determine that sufficient data is

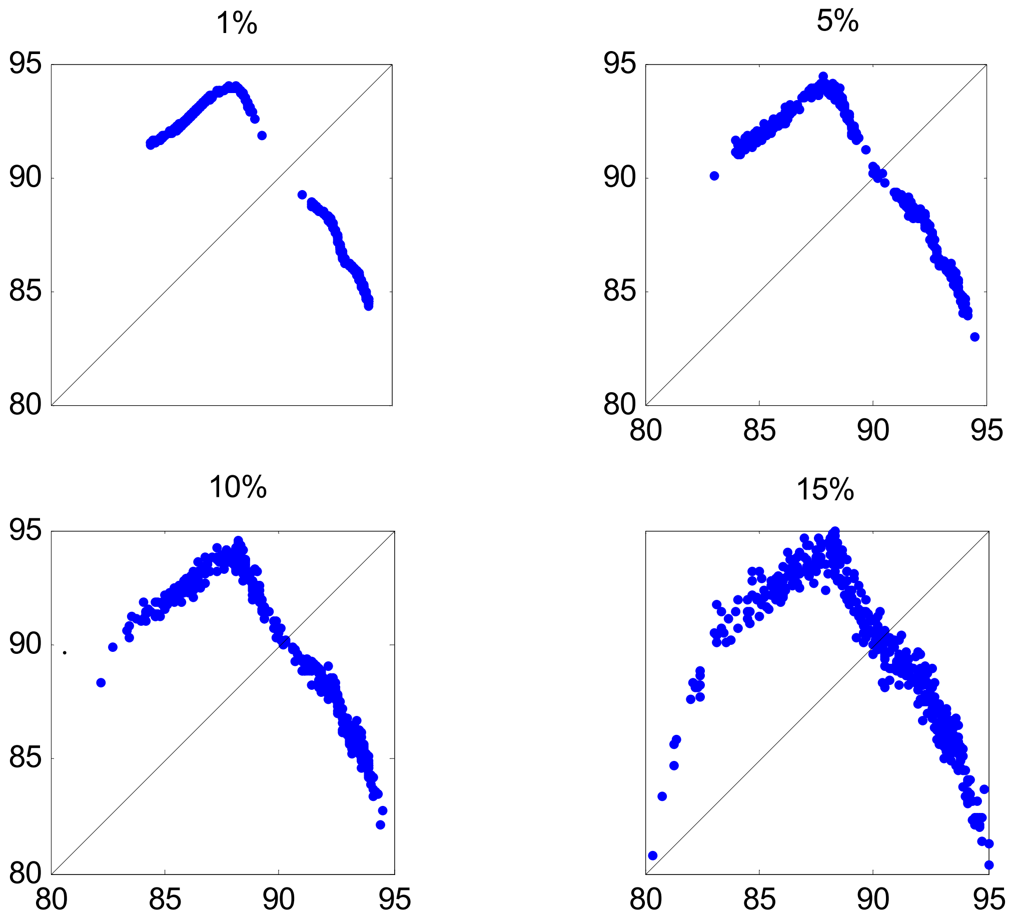


Figure 4.14. Iterated return maps for various amounts of noise

Clockwise from top left: Noise added to the iterative step 1% (a) , 5% (b), 10%(c) and 20% (d) noise respectively. The noise was added in the form *t(i+1)=f(t(i))+*e_i where e_i~N(0,(cσ)^2) where σ is the standard deviation of the measurements {***t(i)***}, and *c* is the noise level (0.01, 0.05, 0.10 and 0.15 in (a) through (d)) present for fitting a CLPC. Of course, if there is only a limited amount of data, the number of cluster centers can be increased to improve the fit, but that cannot deal with the problem posed by severely non-uniform distribution. This aspect about CLPC was discussed earlier, and any algorithm based on conditional expectation will suffer from this weakness. In general, though, it is doubtful that any algorithm can attempt to explain the structure given no data!

4.6. Cluster-linked principal curves in delay space

Principal curves can be used to approximate the reconstructed geometry of the attractor. Figure 4.15 shows the embedding space constructed from a chaotic bubble column time series. See figure 3.3(d) for the time series from which the embedding was produced. Since the data is very clean, 5% white noise[11] was added to it for robustness of estimation. The simulation used 45 cluster centers. An embedding dimension of 3 and an embedding delay of 15 were chosen to resolve its geometry or to 'open up' the attractor. The data points appear as dots and the principal curve as the solid line. It is clear that the principal curve approximates the geometry excellently. The curve is non-intersecting when seen in any two dimensions.

The first two dimensions of the residuals are shown in figure 4.16. The fraction of generalized variance remaining in the residuals is less than 0.07%. The mean of the residuals is very close to zero, and the measure suggested for independence of residuals is 1.041. Figure 4.16 clearly shows how random-like the residuals are.

[11] The noise was added in the form $t(i+1)=f(t(i))+e_i$ where e_i~N$(0,(0.05\sigma)^2)$ where σ is the standard deviation of the measurements $\{t(i)\}$.

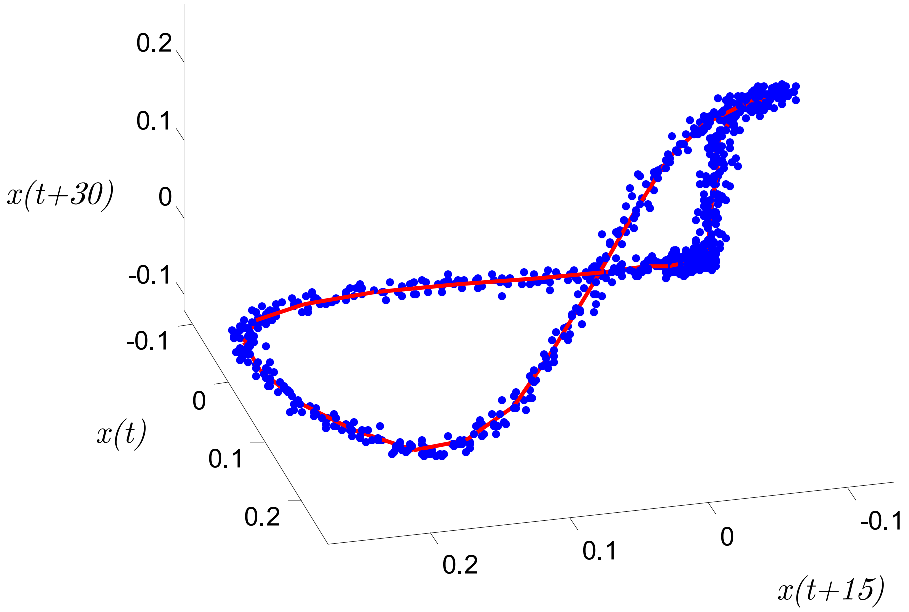


Figure 4.15. Principal curve fitted to embedding space

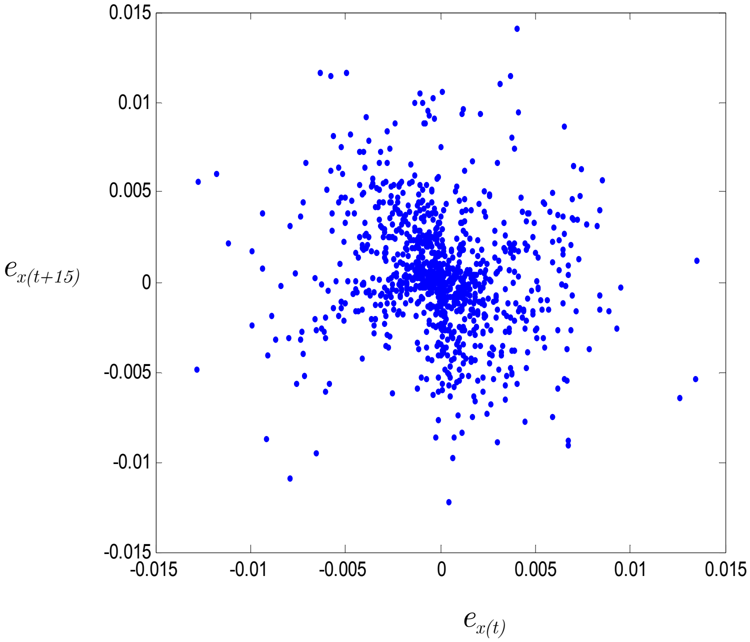


Figure 4.16. Residuals from the fit in figure 4.15 (first two dimensions)

This example shows that principal curves can be used to model the general shape of the state space trajectories in the embedding space. The smallest embedding dimension required for this data set is three, but since the fit produced white noise residuals, the three dimensions can be simply represented by the arc length along the principal curve.

To expand on this point further, observe figure 4.17, which displays the arc lengths for the embedding vectors formed so that the time interval between every record (or embedding vector) is the same. Every arc length is the projection of a 3-dimensional embedding containing a time window of 30 records on the fitted CLPC.

It was noted earlier how the differential pressure between the nozzle and the gas intake pressure builds up slowly but sharply declines when the bubble is released. This pattern is present in the time series as well. The temporal properties of the time series are preserved remarkably well. The sharp increase and decline in the arc length is representative of the same pattern in the time series. This projection of the embedding on the CLPC clearly contains more information that the original time series, and can be employed for visual examination or subjected to standard Statistical Process Control (SPC) techniques.

It must be noted that the principal curve approximated very well in the higher- and lower- density regions. This delay space corresponded to a chaotic state where the trajectories were contained in a band. It is not expected that CLPC or such methods can approximate the complex, fractal nature of strange attractors. Instead the intent is to demonstrate that this method can be used to approximate the general distribution or a skeleton of data points.

The next chapter concerns itself with the applications of Cluster-linked Principal Curves. The distribution of arc lengths obtained from return

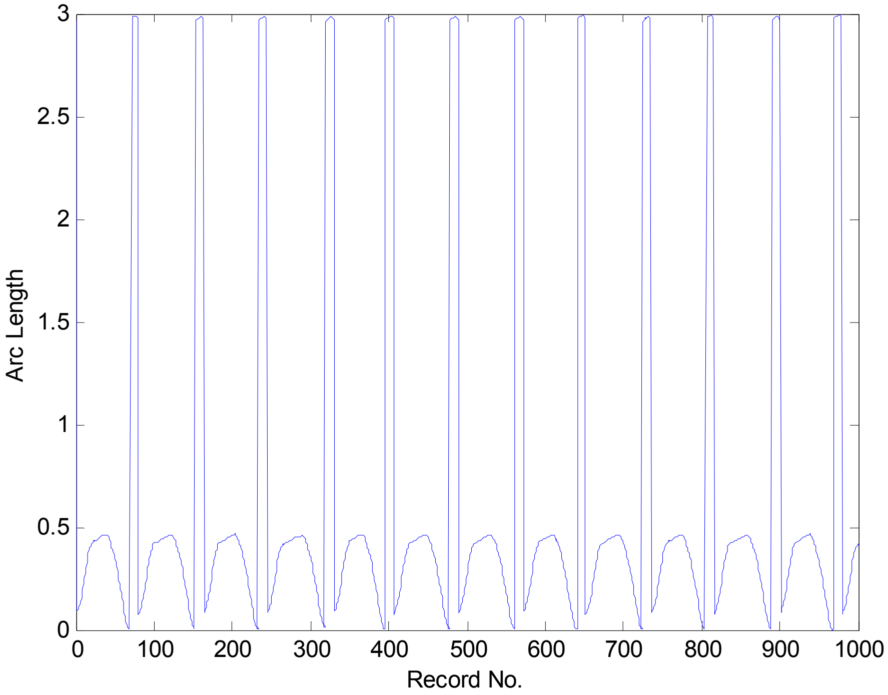


Figure 4.17. Arc lengths for the data in figure 4.15

The CLPC used to obtain these arc lengths is shown in figure 4.15

maps or embedded times series is used to test for reversibility and stationarity. The extension of this methodology to process monitoring and fault diagnosis is outlined. The extension to prediction is also briefly discussed.

4.7. Remarks about cluster-linked principal curves

Some remarks are in order about the algorithm presented and the examples presented. First of all, the principal curves are weak approximators at their extremities. The tails of the arc length distribution produced by projecting on a principal curve are not very reliable. Using finer approximation at their extremities leads the danger of the CLPC being too sensitive to even small statistical fluctuations near its endpoints. This high sensitivity may also be passed to the neighboring areas and mar the fit of the entire principal curve.

The second remark has to do with the validity of a principal curve when it is faced with sparse data. If the data being approximated is sparse, and a large part of the principal curve has no data points near it –and is simply a product of connecting one cluster of data points to another, then one must ask oneself if the probability of points in that region is very small or zero. Bayesian analysis, for example, doesn't assume any probability is zero, but expects some finite, however small, probability even in the regions where there is no data. It has already been observed that the conditional probability is not well defined for very low probabilities. In some cases, it may be an artifact of the data that a considerable region in the data space contains no data points. For engineering systems, e.g., it is possible if the system is operated, at two different steady states, in such a way that the transition is almost instantaneous in relation to the sampling frequency.

If the data contain two compact clusters far apart, and the user is aware of it, the CLPC algorithm should not be considered reliable in its interpolation. For such cases, other methods can be used.

That also brings us to the problem of determining how accurate the probability distribution of arc lengths really is. Kernel smoothers may be used to better estimate the probability of observing a certain arc length. However, their use cannot hide the fact that there are regions in the data where there are no observations.

No simple answer can be given to these questions. If, after binning the arc lengths to produce their probability distribution function, a large number of congruent bins are empty, a note shall be made of it. The user can determine for himself or herself whether the result is reasonable or not. The CLPC algorithm does produce some principal curves that intersect themselves –and may give rise to a curve which finds no observations for a considerably large part of its traverse. The range of arc lengths resulting from such a CLPC will be larger. A rough idea of the range of the arc lengths can be obtained from the variance of the data. If the range of curve arc lengths is 10 times the generalized standard deviation, further inquiry must be made into the shape of the fitted curve.

This dissertation is concerned with demonstrating how the CLPC algorithm can be put to various uses. We wish to state here that we are aware of some questions, mostly statistical in nature, which are beyond the scope of this study. At the same time, most of the questions posed above are not endemic to the CLPC algorithm, but are universal when analyzing data from an unknown source.

Half of this game is ninety percent mental-

Yogi Berra

Chapter 5

Applications of cluster-linked principal curves

This chapter shows how cluster-linked principal curves can be used to test for stationarity and reversibility in a time series. Chapter 4 demonstrated how principal curves can approximate the distribution of data points in return maps as well as in embedding space. This chapter shows some additional results with the arc length distributions along the CLPC.

5.1 Comparing two distributions

A simple way to test for stationarity in univariate data sets is the F-test, which tests if two samples have unequal variances. The test is based on the assumption that the two samples whose variances are being compared are random variables. It can be applied to a time series or a one-dimensional distribution as Goldfeld-Quandt (G-Q) test to confirm heteroscedasticity (unequal variances). G-Q test requires computing the variances of the first and last one-third of the data. The idea is to produce two equal-sized non-overlapping segments that are considerable removed from each other. Using the first and last one-thirds appears to be a popular choice. If the measurements are normally distributed, the ratio of variances follows a F-statistic.

If there are n samples overall, then the ratio of the variances of the first and last one-third follows the F statistic with $n/3$ degrees of freedom for numerator as well as denominator. A simple F-test can then determine if the variances are significantly different. If they are, the time series is considered non-stationary. Note that the F-test is parametric and assumes that the measurements are independent, which is not the case for most time series. However, a static probability density function may be sufficient to characterize a time series if enough measurements are collected so that the density estimated from the sample is representative of the probability density function of the generating process.

Principal curves can also be used in the same fashion to test for stationarity. A principal curve can be fitted to the first one-third of the data and yields a distribution of the arc lengths for it. Then the arc lengths for the last one-third of the data can be found by projecting it on the principal curve fitted to the first one-third of the data. Then one has two distributions of arc lengths and comparing them is tantamount to evaluating the *goodness of fit*. The Kolmogorov-Smirnov[12] (K-S) test is the standard non-parametric test to compare two distributions if they are continuous function of one parameter[13]. However, the K-S test is not powerful for distributions with long and weak tails since the difference between the density functions being compared will be small and not appear to be of much importance[14]. The K-S test statistic is

[12] The Kolmogorov-Smirnov test does not make any assumptions about the distribution. It compares two cumulative distribution functions that are constructed from the data and not according to the quantiles of a parametric distribution.

[13] Strictly speaking, the distribution of arc lengths is not continuous due to finite sample size.

[14] The K-S test uses the infinite norm of the difference between two cumulative distribution functions as a test statistic. It is much more sensitive near the center of the distribution than it is far from it.

the infinite norm of the difference between two cumulative distribution functions $F_R(X)$ and $F_S(X)$ where the subscripts index the distribution functions. Note that X must be a continuous variable for the K-S test to be valid.

The χ^2 test on the other hand, accumulates the differences in the densities into one statistic. Nevertheless, using the χ^2 test requires one to specify the number of bins in which the densities are binned. There are ways to decide on the optimum number of bins by using informational complexity measures, which is beyond the scope of this study. We arbitrarily used $n/20$ bins where n is the number of observations in each sample[15]. However, no less than 30 bins are used when n is less than 600. The goal of this study is to show how the distributions of arc lengths along the CLPC can be used to characterize a time series and the results are only illustrative. It is not advocated to blindly set an α-value and to accept or reject hypothesis based on the fixed cutoff. The associated p -value from the comparison of the densities should only be used as an aid. For comparison, the results of the Kolmogorov-Smirnov (K-S) test are provided. For details on these two goodness of fit texts, see Press et al (1993), or any standard Statistics text.

This approach presents a more efficient way to test for stationarity if the principal curve explained most of the variation in the data set *and* if the residuals produced by it were white noise. The argument is that the principal curve extracted most of the information present in the distribution and thus

[15] If there is no natural choice for the number of bins, we suggest that the chi-square test be carried out for various value of N_B. If the difference is significant, the test will reject the null for all values of N_B. It must be borne in mind though that using too few bins may ignore a difference between the distributions. Results obtained with too few bins should not be given much weight.

the segments of the time series can be compared in one dimension instead of their original, higher dimension. It is also true that comparing higher dimensional probability distributions is a cumbersome task and is still a subject of ongoing research. The chi-square test is defined as follows:

Suppose R and S are two binned probability distributions, and R_i and S_i are the number of samples in their i^{th} bin respectively. If both distributions are determined experimentally, then the appropriate chi-square statistic to test for the distributions being significantly different is

$$\chi^2 = \sum_{i=1}^{N_B} \frac{(R_i - S_i)^2}{(R_i + S_i)} \tag{30}$$

where N_B is the number of bins.■

The statistic follows the chi-square distribution with N_B-*1* degrees of freedom[16]. The null hypothesis of R_i and S_i being the same can be tested and if the χ^2 value is larger than the critical χ^2 value at a defined confidence level, the null can be rejected and it can be concluded that R_i and S_i are significantly different and thus the time series or the dynamics are not stationary. As usual, the inability to reject the null hypothesis doesn't prove that the time series is stationary.

We now demonstrate the uses of arc length distributions by testing for stationarity and reversibility based on return maps and delay embedding. The time series used for illustration are taken from a liquid-filled column with electrified capillary (or a bubble column). A brief description the bubble column is given in section 3.1 of this dissertation.

[16] Not that the degrees of freedom are N_B-1 if the bins are considered to be independent.

5.2 Testing for stationarity

Figure 5.1 shows the overlaid return maps formed from the first and last one-thirds of the time series. The blue dots correspond to the first one-third of the series and the red dots to the last one-third of the series. It is clear that the distribution of points is different for the first and last one-thirds of the data points. A CLPC with 16 cluster centers was fitted to the blue dots, and used to obtain the arc length distribution for the red dots.

The overlaid binned probability distributions are shown in figure 5.2 where the dashed and solid lines show respectively the distribution of arc lengths for the last and first one-thirds of data points. The distributions present some interesting features. The dashed density has four distinct peaks –thus attesting to the period-4 behavior of the red dots. The solid line has a distinctly different structure. Thirty bins were used to generate figure 5.2, and the null hypothesis was rejected at α<0.01 or greater than 99% confidence level. The corresponding p-value was 5x10^{-7}. Using 50 bins, the corresponding *p*-value was 8x10^{-6}. In other words, the probability that the two segments have the same distribution is less than 1% under the null. The *p*-value for the K-S test is 0.056, which is fairly small. It is thus quite likely that the time series is non-stationary.

The distribution of arc lengths can also be used to find the periodicity in the data. For example, two clear peaks signify period-2 behavior, four sharp peaks signify period-4 behavior and a broad distribution signifies chaotic or random behavior. Note that while fitting the principal curve the return map data points were scaled to zero mean and unit variance, and that is why the arc lengths range from 0 to 4.

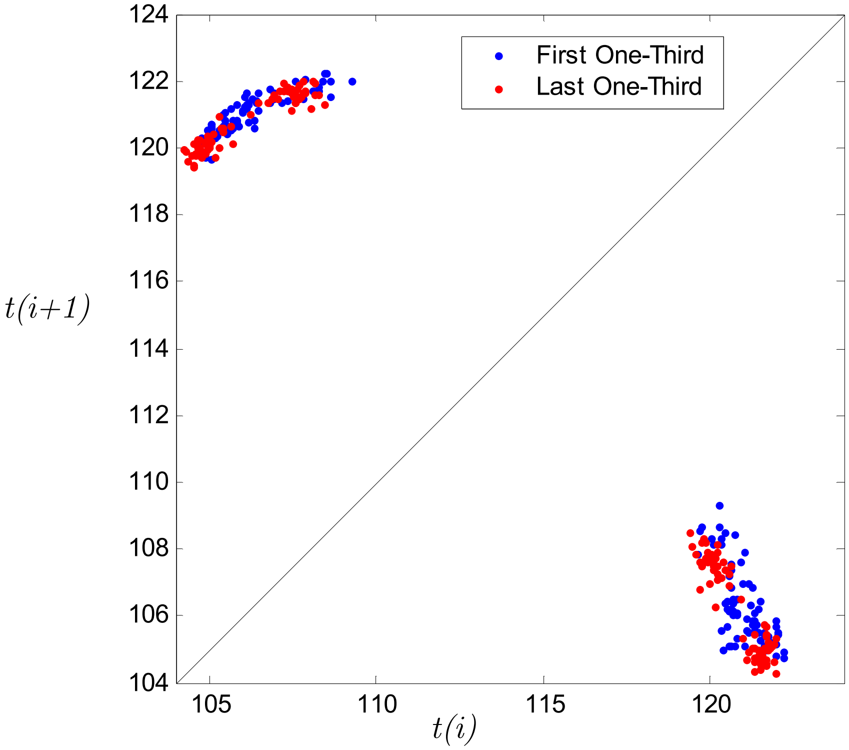


Figure 5.1. Overlaid return maps for a bubble column time series I

First one-third of mean crossings are plotted in blue, and the last one-third are plotted in red. The time series was obtained from the bubble column with gas flow rate of 170 cc/min and the electrostatic potential across the electrified capillary was 12000 V. The units for *t(i)* are milliseconds.

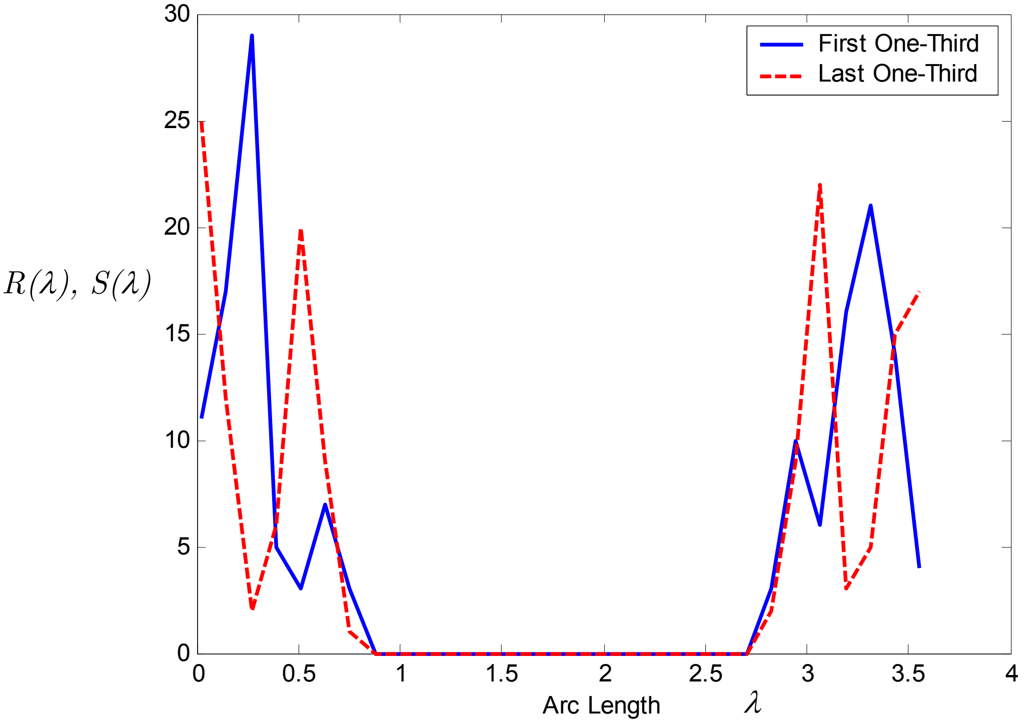


Figure 5.2. Distributions of arc lengths for the data in figure 5.1

We now show another, example based on a bubble column time series. Here the objective is to see if a chaotic time series is non-stationary. Figure 5.3 shows the overlaid return maps from the first and last one-thirds of the time series. Once again, blue corresponds to the first one-third and red to the last one-third of data. Apparently the distributions are not very different. However, note that the distribution of the red dots is more uniform.

This is quite noticeable at the top left and bottom right as well as near the diagonal. A principal curve was fitted with 15 cluster centers for the mean crossings of the first one-third of the time series. Then the mean crossings for the last one-third of the data were projected upon it to produce another distribution. Figure 5.4 shows the overlaid probability distributions of the arc lengths. 30 bins were used to produce figure 5.4.

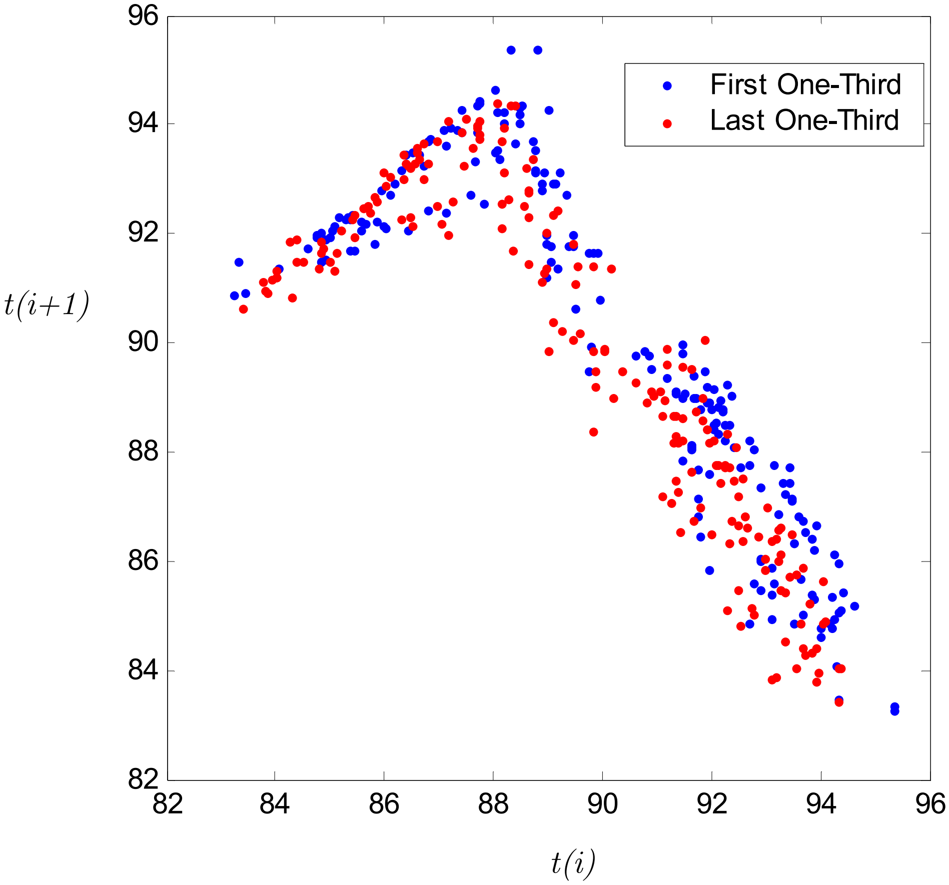


Figure 5.3. Overlaid return maps for a bubble column time series II

First one-third of mean crossings are plotted in blue, and the last one-third are plotted in red. The time series was obtained from the bubble column with gas flow rate of 170 cc/min and electrostatic potential (across the electrified capillary) of 17000 V. The units for abscissa and ordinate are milliseconds.

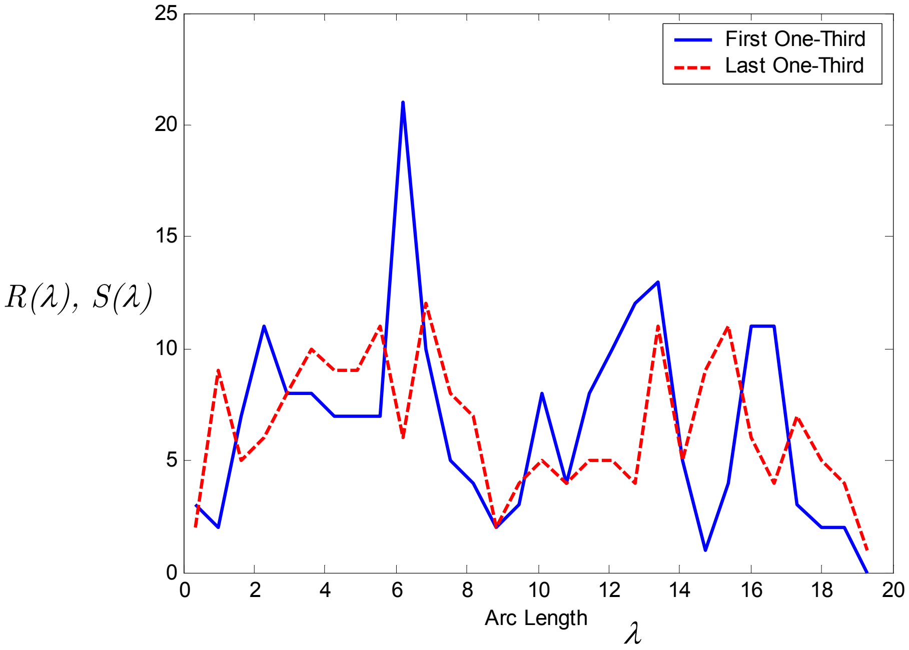


Figure 5.4. Distribution of arc lengths for the data in figure 5.3

The distributions look different but not very much so. However, the null was rejected at α<0.01 for 30 and 50 bins. The corresponding p-values were 0.0074 and 0.0017 respectively. This is a good illustration of the weakness of K-S test. The K-S test did not reject the null of stationarity, and the corresponding p-value was 0.49. It is easy to see in figure 5.4 that the cumulative distribution functions will not be very dissimilar. For comparison, see figure 5.5 which shows cumulative distribution functions for the first and last one-thirds of CLPC scores.

The cumulative distributions do not look very different, and that is why K-S test didn't reject the null hypothesis of stationarity. The difference between the return maps 5.3 is not very obvious. However, it is likely that the series is not non-stationary and the K-S test gave the 'correct' result. This example also demonstrates the subjectivity of hypothesis testing.

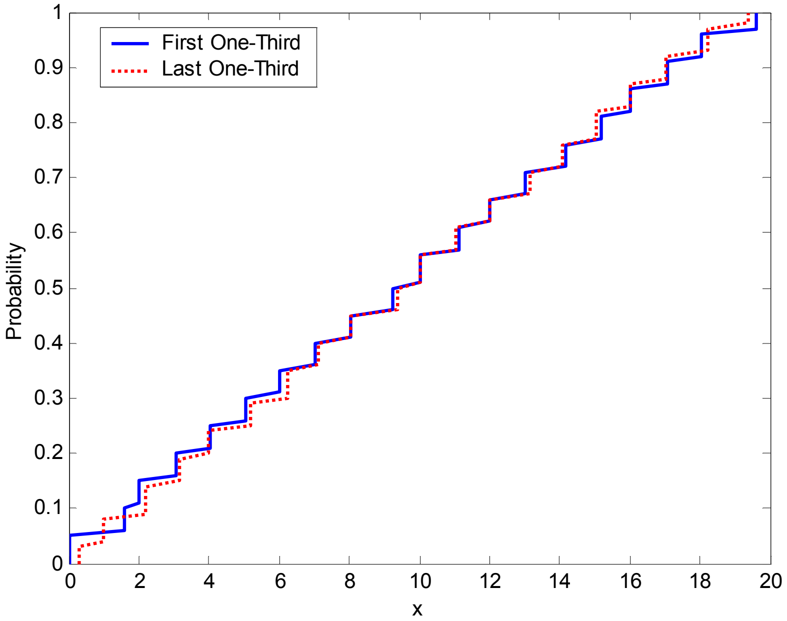


Figure 5.5. Distribution functions for the distributions in figure 5.4.

The next example shows how one can test for stationarity based on the delay embeddings. The time series used in this study was taken from a bubble column under chaotic operating conditions. The embedding dimension used was three, which is the minimum dimension to reproduce the geometry of the bubble column attractor and to observe no intersecting trajectories. Note that halfway through the experiment, the operating conditions were altered slightly. The system was allowed to settle before resuming measurements, and thus it is known that this time series is non-stationary.

Figure 5.6 shows the overlaid reconstructed attractors in embedding space[17]. The blue and black dots correspond to the first and last one-thirds of the series respectively. Figure 5.7 shows the overlaid arc length distributions for the first and last one-thirds of the data set. The distributions are clearly

[17] In figure 5.6, the mean of the embedding formed from the first one-third of the time series was subtracted from those formed from the first and last one-thirds of the time series.

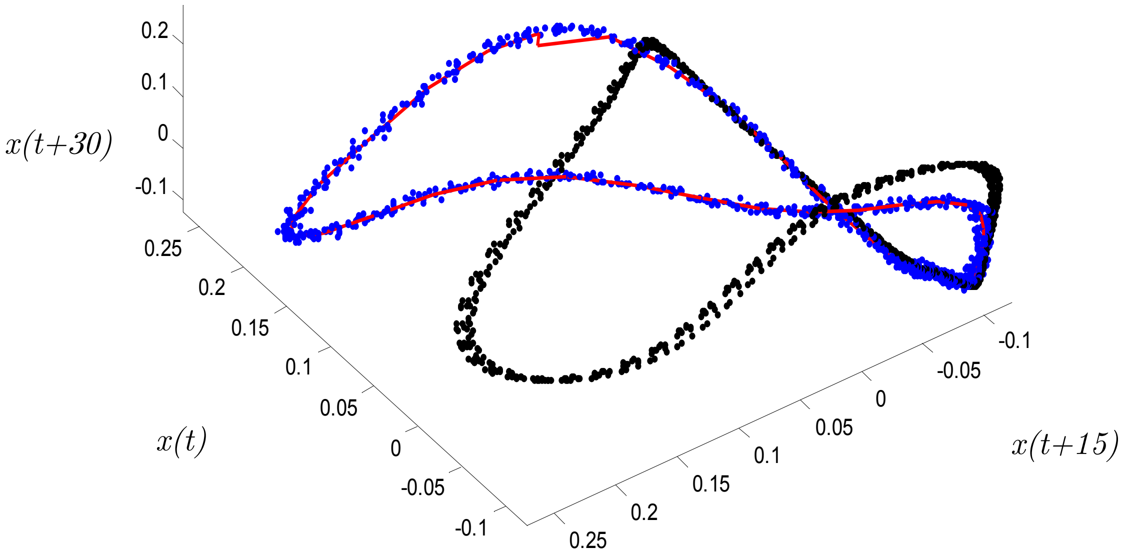


Figure 5.6. Overlaid attractors in embedding space I

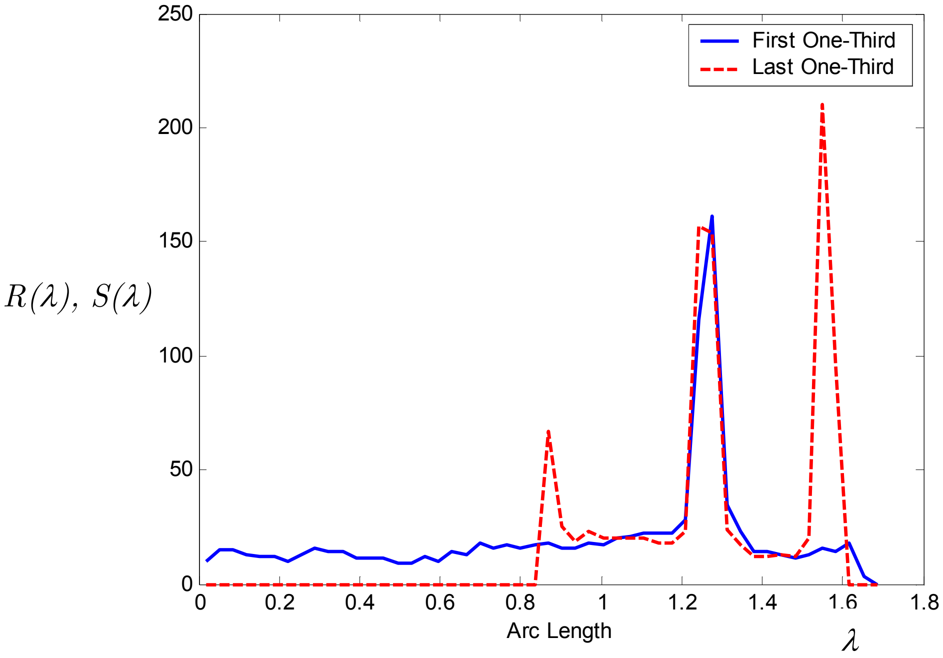


Figure 5.7. Overlaid PDFs for the data in figure 5.6

very different. For the distributions in figure 5.7, the null hypothesis that the two arc length distributions are generated from the same distribution function was strongly rejected. In fact the associated *p*-value was 4×10^{-68} for the K-S test and even less for the chi-square test, which leaves no doubt about the fact that these two distributions are *very* different and hence the time series is non-stationary when seen in the embedding space.

The final example of this section tests stationarity for another time series based on the reconstructed attractor. The operating conditions were unchanged during the period the time series was obtained. Figure 5.8 shows the embedding made from the first and last one-thirds of the series (blue and black dots respectively). There does not seem to be a systematic difference between the two reconstructed attractors, as is clear from the figure.

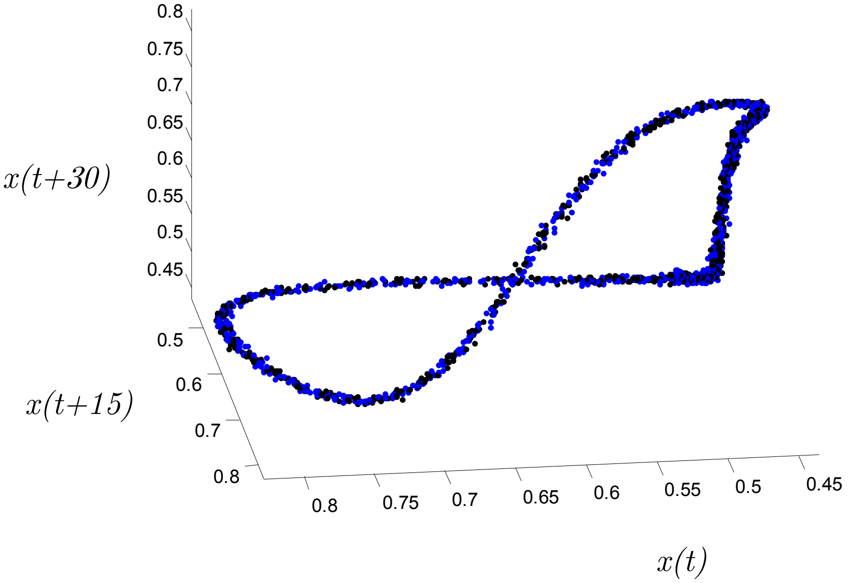


Figure 5.8. Overlaid attractors in embedding space II

The principal curve fit to the first one-third of the time series is not shown, but it was quite similar to the one in figure 4.15. The principal curve began and ended in the high-density segment of the embedding space. The high-density and high-probability zone corresponds to the formation and expansion of the bubble, and the lower density zones correspond to the detaching of the bubble from the nozzle. The probability distributions are shown in figure 5.9 and almost lie atop each other. As is obvious, the null hypothesis of stationarity was not rejected by chi-square or K-S test at p-value of even 0.25. Another stationarity test based on the return maps also upheld stationarity. Thus there is strong evidence that this time series is stationary.

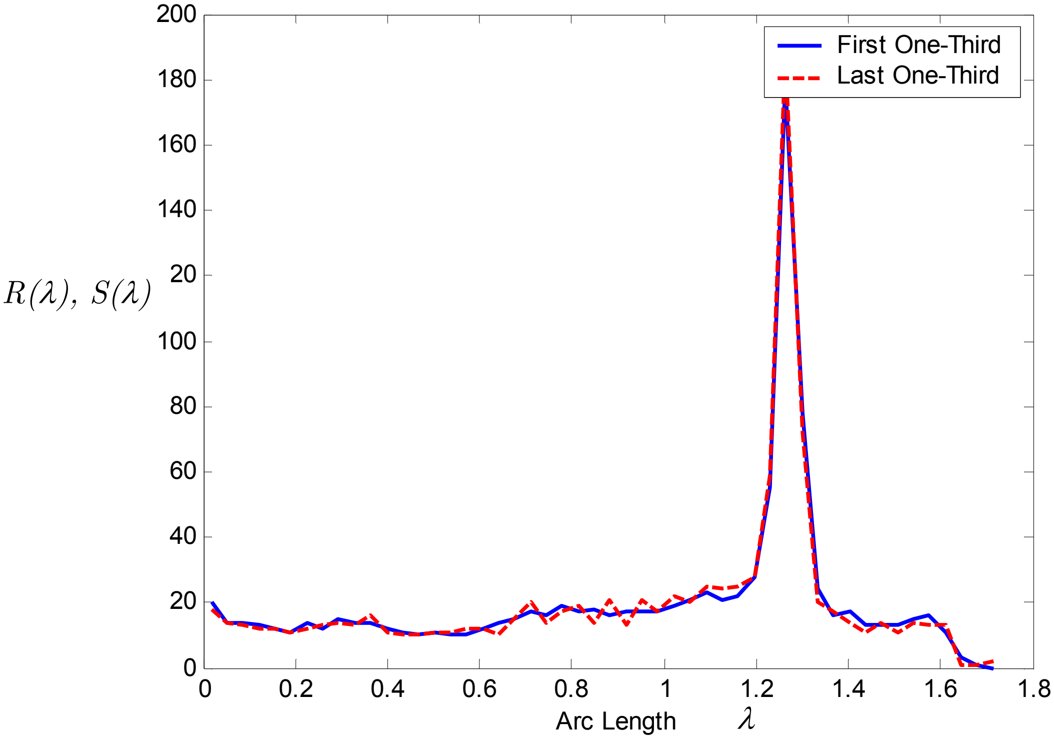


Figure 5.9. Overlaid PDFs for the data in figure 5.8

This section contained four examples of testing for stationarity using Principal Curves. These examples covered respectively, high, medium, high and little non-stationarity in the data. The first two examples used the return maps whilst the last two employed the embedding space directly. In the first example the return maps were different and it was accurately captured. In the second example, the difference was found to be significant by the chi-square test but the *p*-value from the K-S test was not low enough to reject the null of stationarity. However, the return maps were somewhat different, but not very much so. A ratio of the computed chi-square statistic to the critical chi-square value can be used as an indicator of stationarity along with the *p*-value from the K-S test. In the third example, there was a big difference between the distributions of the first and last one-thirds of the data, and the null hypothesis of stationarity was rejected very strongly by chi-square as well as K-S test. And, in the final example, there was little difference between the distributions of the first and last one-thirds of the data and neither test indicated a low enough *p*-value to suggest that the null may be unlikely.

The preceding examples show that the approach outlined in this section is effective and doesn't suffer from the problems of being too powerful or too weak. It must be borne in mind that the examples considered here didn't have a lot of data, and the accuracy of the test will improve as more data becomes available to estimate the probability distributions of arc lengths. In particular the chi-square test becomes increasingly powerful with sample size. If there is enough data, it is preferred to have tests based on the cumulative distributions or normalized distributions.

This approach can also be used to compare two different processes. Obviously if the two processes have a different mean, one does not have to go through the trouble of fitting a principal curve when a *t*-Test or Hotelling's T^2 statistic can determine that the means are significantly different. The above approach can still be used to compare the structure of data points of two different data sets. Both time series can be normalized so that they have zero mean and unit variance. Then the scaled data points will occupy roughly the same region. A principal curve can then be fitted to one return map and the arc lengths on the resulting principal curve can be computed for another return map. For more examples see Rajput and Bruns (2001A).

5.3. Testing for reversibility[18]

Most nonlinear time series are irreversible. The knowledge of reversibility of a time series is very useful since some models can be ruled out for irreversible time series. The approach taken in the previous section can be modified slightly to test for reversibility in place of stationarity. Instead of comparing the first and last one-thirds of a time series, one can compare the time-forward and time-reverse versions of the series[19].

One can fit a principal curve to the return map formed from the forward version of the time series. Then the fitted principal curve can be used to find the arc length distributions for the return map formed from the time-reversed version of the time series. The two resulting distributions can

[18] Note that one should only test for irreversibility if the time series is known to be stationary. Non-stationary time series are by definition irreversible.

[19] The same methodology applies whether we compare a time series with its time-reversed version in the embedding space or compare the return map formed by mean crossing intervals to the return map formed by mean crossing intervals of its time-reversed version.

then be compared in the same fashion using either a chi-square test or a K-S test.

Figure 5.10 shows the overlaid return maps from a bubble column time series. The same time series was used to test for stationarity in figure 5.1. The return map of the time-reversed version appears as red dots and that of the time-forward version is shown in blue dots. Clearly the time series is not reversible since the red and blue dots do not overshadow each other in the figure 5.10. However, the irreversibility is not very strong, since the return maps of the time-forward and time-reversed time series overlap considerably.

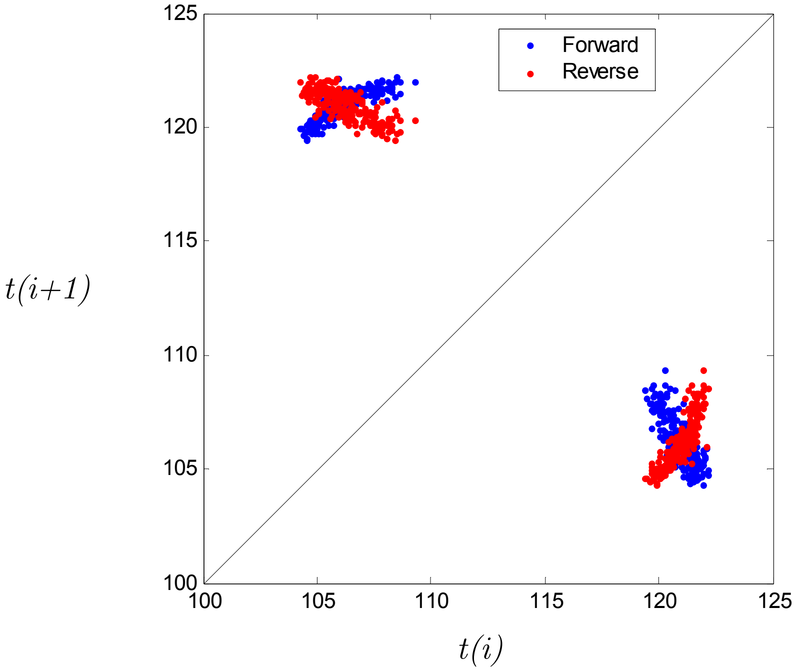


Figure 5.10. Return maps for time-forward and time-reverse versions of a time series

The units for abscissa and ordinate are milliseconds.

A principal curve was fitted to the time-forward version of the time series and the points in the return map of the time-reversed version were projected on it, and another probability distribution obtained. Figure 5.11 shows the overlaid probability histograms for the forward and reversed versions of the time series.

The differences seen in figure 5.10 are visible in figure 5.11 as well. The null hypothesis of the distributions not being significantly different was rejected very strongly. The corresponding *p*-values for 30 and 50 bins respectively were 4 x 10^{-8} and 4 x 10^{-5}. The K-S test also rejected the null of reversibility with an associated *p*-value of 0.005. Therefore there is very strong statistical evidence that the time series is irreversible. This time series is thus non-stationary and irreversible, which comes as no surprise. However,

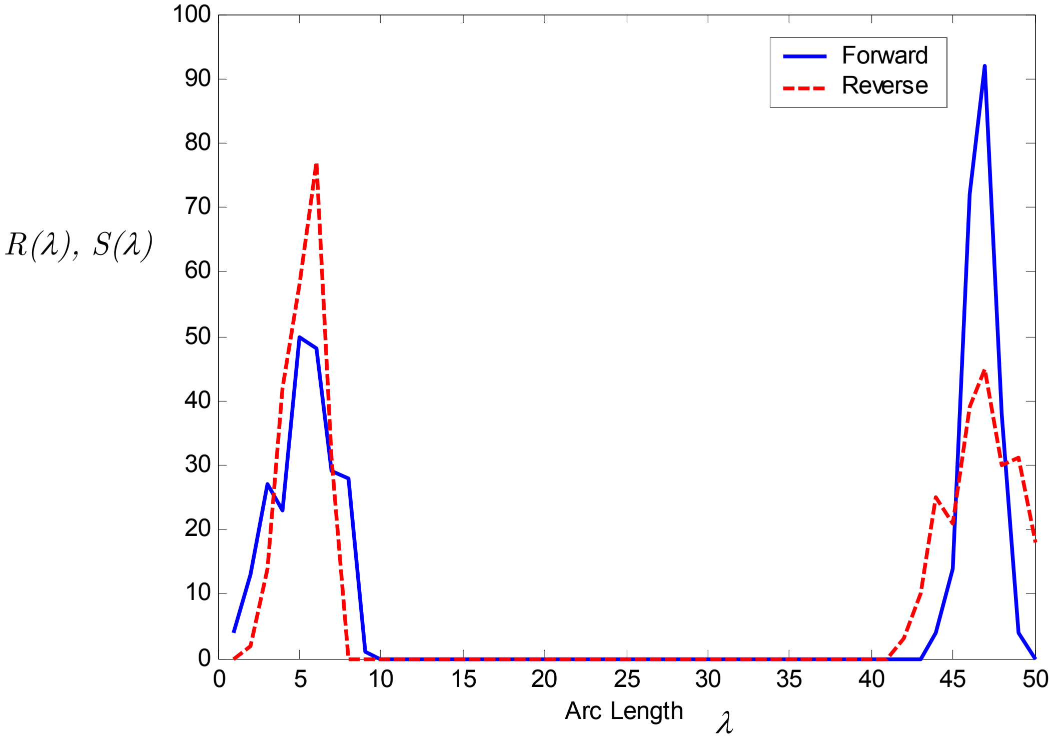


Figure 5.11. Overlaid probability distributions for the data in figure 5.10

the very low p-values associated with these tests assure us that this methodology finds non-stationary time series irreversible.

Figure 5.12 demonstrates how principal curves can be used to test for reversibility based on the delay embeddings. This example considers the same data that was used in figures 4.15 and 5.8 –that of a stationary time series from a bubble column operating under chaotic conditions. As before, a principal curve was fitted to the time-forward version of the time series and later the time-reversed version of the same time series was projected on the principal curve to obtain another probability distribution of arc lengths.

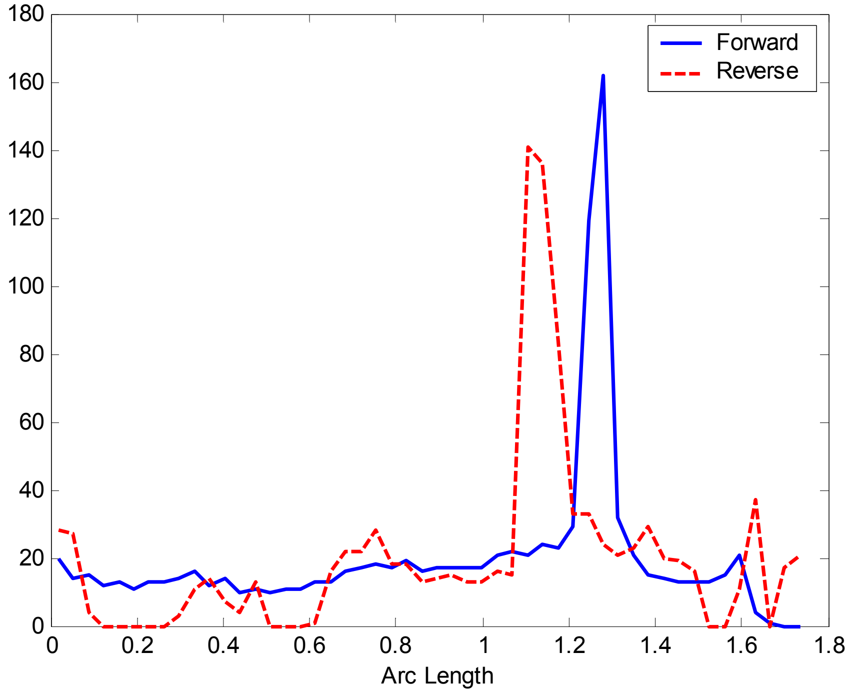


Figure 5.12. Overlaid probability distributions for the time-forward and time-reverse versions of a bubble column time series

It is obvious that the two distributions are widely different. The null hypothesis of reversibility was rejected very powerfully. The chi-square test produced a *p*-value of 4×10^{-8} and 5×10^{-5} for 30 and 50 bins respectively. The K-S test also rejected the null since the associated *p*-value was less than 10^{-10}.

So it can be safely concluded that the time series is irreversible. It was expected though, because the time series was chaotic and chaotic systems are not time-reversible.

5.4. Process monitoring and fault diagnosis

The methodology discussed in the last two sections can be applied to monitoring. There are two ways in which process monitoring can be approached, and they are suitable for different ends. The usual case is where the *good* operating condition is known and it is desired that the system remains near it. In that case, a principal curve can be fitted to the *good* data, and then used to project the data from a running window, to produce the distribution of the moving window. The distribution of the moving window can then be compared with that of the good data. A running ratio of the test statistic to the critical value of the statistic according to the null, and the probability value corresponding to the computed statistic can then be displayed and utilized for monitoring. As observed earlier, the distributional properties of the arc lengths should be explored before deciding whether to have a parametric test or a non-parametric one. It is preferred to have a non-parametric test to make the monitoring more robust.

If the system is complex, then one can follow another approach that in spirit is more like identification. Data can be collected for several operating modes or states, and a library of arc length distributions compiled. Later the

L^1 or L^2 measure of distance between the running distribution and the standard library distributions can be displayed to indicate which mode or state the system is currently in. This approach can be quite useful when there is not a lot of understanding about the process and there are lots of *gray areas*.

The second approach can also be applied to fault diagnosis purposes when a library of system states or states with certain known faults has been assembled. A measure of similarity between the arc length distribution in the present time-window and those in the library can be computed online. When the dynamic, online measure of similarity indicates that the system is in or is moving towards a known fault, a warning is issued. The knowledge of the process can be used to make changes in the system parameters such as flow rates, and thus to better control it. The measure of similarity can be so defined that makes the methodology more conducive to preventive maintenance. That is achieved by issuing a warning for moderate drifts toward known faults. The warning is noted by the maintenance staff. The priorities of routine maintenance may then be optimized by focusing more on the machinery or systems for which repeated and ever-stronger warnings were issued consistently. Thus formulated, the distribution of arc lengths can reduce failures, shutdowns, and maintenance costs.

The HSPC algorithm was extended by Dong and McAvoy (1996) to perform Nonlinear Principal Component Analysis (NLPCA). They followed the HSPC algorithm and trained an autoassociative neural network to learn the mapping from data space to the nonlinear principal component space. Their method fits one HSPC after another (on the residuals of the fit of previous HSPC) till a predefined fraction of total variability is explained. Their article provides examples of process monitoring based on NLPCA.

They follow the HSPC algorithm and fit a scatterplot smoother for every pair of dimensions. The problems with that were discussed in Chapter 4.

The Cluster-linked Principal Curves (CLPC) algorithm is different from the HSPC algorithm or NLPCA approach. The algorithm does not include the cumbersome and black-box-like training of the neural network where there is little idea about the nature of the mapping performed by a neural network. Instead, the data points are projected on the fitted principal curve in real time. The complexity of projection, as noted before, is $O(n*n_c)$ where n is the number of data points and n_c is the number of clusters. Our experience shows that the projection is done very quickly on a 500 MHz machine. Of course a lot has changed from the year 1996 to the year 2002 in terms of computing power, but the limitations of neural networks are numerous and they have been spectacularly abused and applied to situations where a simple statistical technique may have done much better for far less.

The extension of CLPC framework to process monitoring and fault diagnosis is beyond the scope of this study, but it is expected that the extension will be straightforward to implement.

5.5. CLPC framework for prediction

The CLPC framework detailed in this chapter for testing stationarity and reversibility can be extended for prediction. The CLPC reduces a point $\mathbf{x}$ in R^m to an arc length $\lambda(\mathbf{x})$. Consider that y is the future value or the value to be predicted. For embedding $\mathbf{x}=[x(t)\ x(t-\tau)\ \ldots\ x(t-(m-1)\tau)]^T$ and $y=x(t+\tau_2)$. A CLPC can be fitted to the vectors $\{\mathbf{x}\}$, and the arc lengths $\{\lambda(\mathbf{x})\}$ can be computed for every $\mathbf{x}$ in R^m. Suppose that the goal is to predict y_i given $\mathbf{x}_i$.

All the vectors in the neighborhood of $\mathbf{x}_i$ in R^m are then found, i. e. $\{\mathbf{x}_j\}$ such that their corresponding arc lengths $\{\lambda(\mathbf{x}_j)\}$ are close to $\lambda(\mathbf{x}_i)$. The corresponding $\{y_j\}$ values for all these neighbors of $\mathbf{x}_i$ can be averaged or kernel-weighted to produce the predicted value. This approach involves searching for neighbors in R^1 if the first principal curve explained most of the variation, which reduces the cost of neighbor search drastically. To be more meticulous, one could choose the $\{\mathbf{x}_j\}$ such that in addition to having similar arc lengths, they are within a certain neighborhood of $\mathbf{x}_i$ (in R^m), and use that set to predict the future value. Further research is underway on this subject, and is beyond the scope of this study.

5.6. Closing remarks

This chapter showed how the Cluster-linked Principal Curves can be used for various ends, and applied them to chaotic time series data in the embedding space. The CLPC fitted to the embedding space captures only the static probability density. Although embedding incorporates some information in the vectors, and the CLPC has a sense of direction (cf. figure 4.16) and approximately remembers the 'arrow of time' (except at its beginning and end where a discontinuity is possible), the timescales that a CLPC can capture are fixed by the time-window in an embedding vector or the time $(m\text{-}1)\tau$. A suitable embedding delay has to be found before subjecting the embedded vectors to CLPC in order to have a more meaningful skeleton of the attractor. If one desires to explore the timescales in the time series through the CLPC framework, one has to repeatedly fit a CLPC for different embeddings. Given some prior information, the researcher can focus more on the relevant time scales by choosing the appropriate embedding. The CLPC framework outlined in this dissertation proved useful in characterizing the joint probability density of the embedded time series, and shows promise if employed for prediction, but a CLPC essentially captures the static distribution of data points.

It is also very desirable to explore the timescales in the time series. Autocorrelation function is suitable for linear time series but not pertinent to nonlinear time series since it captures only linear temporal relationship. General autocorrelation or mutual information (cf. section 2.5) captures the patterns in the embedded time series. We suggest that using mutual information on symbolized time series (where the embedding vectors are coarse-grained to produce a code) can reveal temporal information. The mutual information function evaluated for various delays provides knowledge about the major time scales in the series, since the mutual information function has local peaks at τ when the knowledge of the measurements at time t adds information about the measurements at time $t+\tau$. The mutual information function can be evaluated at the relevant timescales or delays, and the composite vector can be treated as the raw feature vector in a pattern classification problem. This approach is discussed in the next two chapters.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

–Claude Shannon

Chapter 6

Information-theoretic quantities

This chapter discusses some measures of a time series in the realm of information theory. The most fundamental concept in information theory is that of information in a signal or channel, which is related in a straightforward fashion to the predictability and randomness in them. We will provide a short theoretical description of information theoretical tools that we intend to use, and briefly review how these various measures have been used in the past. The theory is followed by a discussion of symbolization and the effect of symbolization parameters on the computed information theoretical measures. The next chapter deals with applications of the measures discussed here.

6.1. Randomness and entropy

If $\{x(t)\}$ or X is the set of measurements obtained on a channel, then the degree of randomness or lack of predictability of the signal can be quantified as Shannon Entropy. For the original reference see Shannon (1948).

If the measurements obtained from channel X are binned, the value $H(X) = -\sum_{i=1}^{N} p_i \ln p_i$ is called the Shannon Entropy[20]. The subscript p_i is the probability that a measurement falls in the i^{th} bin. N is the number of bins, which are indexed by *i* *(1≤i≤N)*. The logarithm is usually taken on base 2 although that is not necessary. The function H(X) reaches its maximum value of *ln (N)* when all the p_i values are equal, and that is when the signal is the most random, or, extending the deduction by making some assumptions, it translates to the fact that any value is equally likely to be observed. The function attains its minimum value of zero when the probability is zero for all bins except one which occurs when only one value is observed. The definition used in this study is the one similar in spirit to that defined by Tang and Tracy (1998) where $H_S(X) = -\frac{\sum_{i=1}^{N} p_i \; ln \; p_i}{ln \; N}$. The advantage of using this definition is that the maximum value of $H_S(X)$ is 1. A variation of this definition is used in Daw et al (1998) where *N* is replaced by the number of non-empty bins. The definition of Shannon entropy considered in this dissertation is in accordance with that of Daw et al (1998).

[20] Strictly speaking, the entropy is defined only if *x(t)* is discrete. Examples include {Red Blue Black} for the experiment of drawing balls from an urn, or the set of alphabet letters (with the white space) if the experiment consists of transmitting a string of letters from one channel to another. If *x(t)* is more or less continuous, most often some sort of binning is required to keep the total number of bins finite and manageable in computing the entropy. If too many bins are involved, the entropy may be artificially low, because of inadequate sampling.

These two extremes can be thought of as representing the endpoints of the predictability spectrum. Throwing a fair dice results in equal probability of rolling any one of the six numbers on its faces, and the process is most random. In contrast, when $AgNO_3$ is titrated with KCl, one always observes the white precipitate of AgCl without fail.

Shannon entropy attempts to quantify the randomness or predictability in the outcome of an experiment. However, in some cases, one is observing multiple quantities simultaneously and the probability must be defined in a higher dimensional space. An example of the latter is the analysis of embedded time series data for which each embedded point is represented as a vector.

6.1.1. Joint entropy

Let us consider the case for two channels X and Y. In order to analyze the relationship between X and Y, one can compute $p_{i,j}(\mathrm{X,Y})$, which is the probability that a sample from X falls in the i^{th} bin and the corresponding sample from Y falls in the j^{th} bin, then the joint entropy is $H(\mathrm{X,Y}) = -\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p_{i,j} \ln p_{i,j}$. The summation is carried over i $(1 \le i \le N_1)$ and j $(1 \le j \le N_2)$ that index the bins in X and Y. By definition $H(\mathrm{X,Y}) = H(\mathrm{Y,X})$. If the joint process (X,Y) is random, the joint entropy would be high, and if the joint process has some sort of structure, the joint entropy would be less.

6.2. Information-theoretic measures

6.2.1. Mutual information and redundancies

To estimate the additional information obtained about Y by knowledge of X, it is needed to correct for the information contained in X about Y. The additional information provided by the joint distribution of X and Y is the amount by which the sum of entropies of X and Y decreases upon its introduction.

Mutual information measures exactly that by computing I(X;Y)=H(X)+H(Y)-H(X,Y)= $-\sum_{i=1}^{N_1} p_i \, lnp_i - \sum_{j=1}^{N_2} p_j lnp_j + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} p_{i,j} lnp_{i,j}$. See Fraser and Swinney (1986), and Fraser (1989) for details.

If X and Y are independent, then $p_{i,j}=p_i p_j$, and it can be easily shown that I(X,Y) would be zero. The maximum value of mutual information occurs when H(X,Y) attains a value of zero which is only possible if only one bin in the XxY space has all the measurements. In any case, a small value of H(X,Y) which means X provides information about Y and vice-versa, leads to a higher mutual information value. The intuitive idea can be expressed mathematically as I(X;Y)=H(X)+H(Y)-H(X,Y). There is a related concept called conditional entropy which considers the conditional probability $p(i|j)$ which is the probability that a measurement from Y is in bin j, given that the corresponding measurement from X is in the bin i. It is easy to see that $p_i=p(i|j)p_j$. The conditional entropy H(X|Y)=H(X)-I(X;Y)= H(Y)-H(X,Y) describes the reduction in the entropy of Y caused by knowing the joint entropy of X and Y. A recent discussion of mutual information for time-series data is given in Schreiber (2000).

6.2.2. Entropy and mutual information for a time series

Shannon Entropy can be computed for a time series where Y is obtained from X through a delay operator. In that case, $p_{i,j}(\tau)$ is the probability that $x(t)$ falls in the i^{th} bin and $x(t+\tau)$ falls in the j^{th} bin. In that case, the Shannon entropy is the estimate of the predictability of time series on a time scale of τ. If the Shannon entropy is low for a certain value of τ, called lag, then knowing $x(t)$ provides us with some information about $x(t+\tau)$. If knowing $x(t)$ tells us nothing about $x(t+\tau)$, then the entropy is maximum. That is the case if $x(t)$ occupies the i^{th} bin but $x(t+\tau)$ can be in any bin with equal probability.

The entropy of X, as shown before was H(X). The joint entropy is H(X,Y)=H(X,TX) where T is a delay operator. To simplify the notation let H(X,Y) be replaced with $H_\tau(X)$. If the time series has some structure or predictability, then the $\{p_{i,j}(X,TX)\}$ or $\{p_{i,j}(\tau)\}$ should have less disorder than $\{p_i\}$. This additional information can be formulated as $I_\tau(X)=H(X)+H(X)-H_\tau(X)=2H(X)-H_\tau(X)$. This is called mutual information. Assume that {x(t)} is completely independent of $\{x(t+\tau)\}$. In that case $H_\tau(X)=H(X)+H(X)$ and the mutual information is zero. If on the other hand, $\{p_{i,j}(\tau)\}$ has more structure, then $H_\tau(X)$ is small and consequently $I_\tau(X)$ is large. Therefore the mutual information is zero only when a time series is completely random at a timescale τ, and a positive value of mutual information implies some correlation between the measurement at time *t* and that at time $t+\tau$. Also note that the mutual information is equal to Shannon entropy when τ=0. In Our implementation of the mutual information, it is normalized so that its maximum value is 1. This is not necessary, but it is convenient to see the residual entropy and mutual information plots when both of them are scaled to lie between 0 and 1.

Note that mutual information is not directional. The way mutual information has been formulated, H(X,TX)=H(TX,X), and thus it exploits only the static distribution of the probabilities and is invariant under the time-reversal transformation. The mutual information of a time-reversed series is the same as that of the original time series, and thus cannot be used to gauge reversibility.

6.3. Symbolization

The entropies can be computed based on the slope of the correlation sum versus the radius of neighborhood (ε) plot. A straight-line approximation zone has to be found visually and its intercept yields the estimate of entropy. Correlation sums are very sensitive to noise and in many cases there may not be a straight-line part in the C(ε)-ε plot or the zone may be too small and elude detection. In addition, computing the correlation sum is tedious and computationally very intensive.

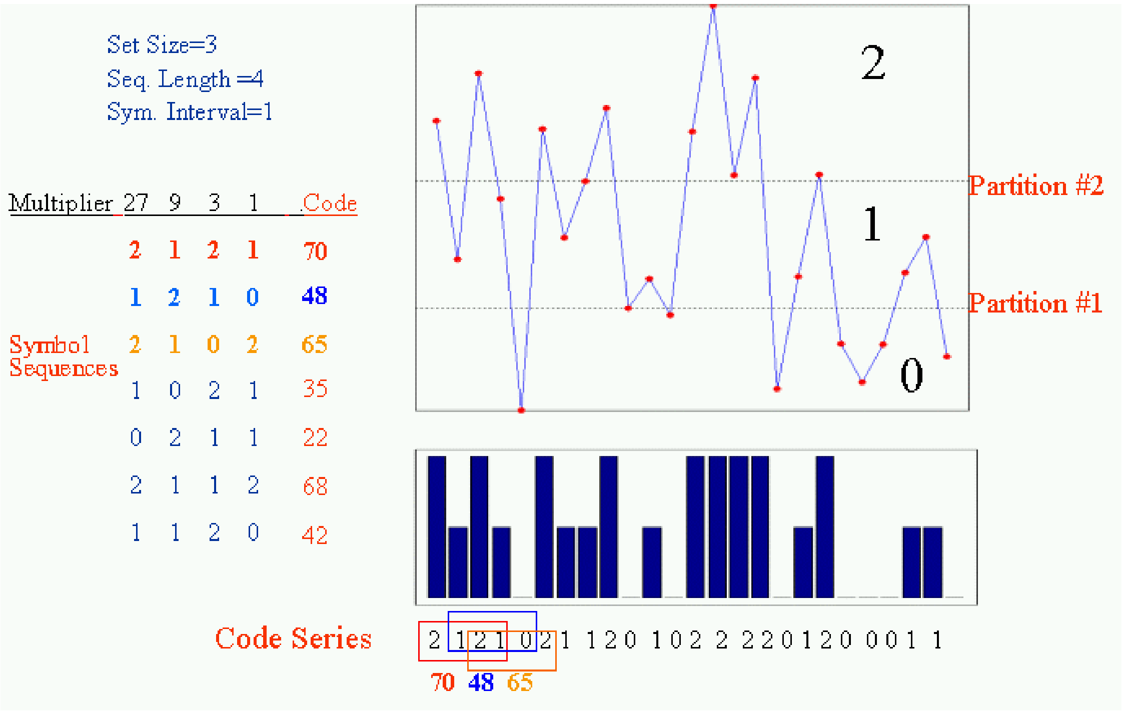
Information theoretic measures motivated in the previous section can be used for computing entropies or other information theoretical measures without computing the correlation sums or finding straight line approximations in the C(ε)-ε plot. Calculation of entropies requires binned probability distribution. Thus one have to decide how finely to partition the data. There are some associated trade-offs in partitioning. Too fine a partitioning scheme produces many bins, of whom most have a negligible probability; too coarse a partitioning scheme leads to loss of information because the large bins group even remotely similar points.

We introduce here the concept of representing measurements by a numeric symbol –0, 1, 2 etc. Our treatment of the subject follows that of Tang and Tracy (1998) in spirit. Early treatment of symbolization was given by Crutchfield and Packard (1983). See Tang and Tracy (1998), Daw et al (2002) and the references therein for more information about symbolization.

Suppose every measurement could be given one of the s values $\{0,\ 1,\ ...,\ s\text{-}1\}$. Assume that the embedding vector is formed from m such symbols. The resulting vector is a s-base number. For ease of representation, it could be converted to a decimal number (called code), although it is not necessary. In this way one obtains a 1-D representation of the dynamics in terms of the code series. Each code thus obtained contains a short history of the evolution of the measurements or the system. Let S be the alphabet size or the set size (which is the cardinality of the set $\{0,\ 1,\ ...,\ s\text{-}1\}$), and m be the symbol sequence length, and by τ the symbolization interval[21]. Figure 6.1 illustrates symbolization.

Of course, not all codes are independent. Of necessity, a code can give rise to only a few codes in the immediate future. A tree can be constructed for better visualization. Consider figure 6.1. The first code is 2121, which

[21] The terminology here is very similar to that described in Chapter 2. The sequence length is plainly the embedding dimension and the symbolization interval the embedding delay. The only new concept here is that of the set size or alphabet size.



Multiplier	27	9	3	1	Code
	2	1	2	1	70
	1	2	1	0	48
Symbol Sequences	2	1	0	2	65
	1	0	2	1	35
	0	2	1	1	22
	2	1	1	2	68
	1	1	2	0	42

Figure 6.1: Illustrating symbolization

corresponds to a decimal number[22] of 70. The next code *has* to be one of 1210, 1211 or 1212. Thus only 3 out of 80 codes are possible for the next code. Because the set size is 4, the fourth next code will be statistically independent of the current code. The probability of generation of a code at any particular instant is not completely random. However, the lack of complete randomness is not important because the goal is not to treat the code series as another time series whose values are random samplings from s. Note that the coding[23] step does not preserve the distance metric.

6.4. Choosing symbolization parameters

As noted earlier, symbolization is affected by three parameters –set size or alphabet size, sequence length and symbolization interval. The second and third parameters can be determined in the same way as embedding parameters. The sequence length and symbolization interval should be so chosen that a time-window described by a sequence is neither too small nor too large with respect to the dynamics of the process. Deciding upon a suitable set size is not obvious, and depends on the complexity of the time series under study. If the process complexity is very high, finer partitions may be required to better capture the various patterns. On the other hand, if the patterns in the system are simple and few, a smaller set size should be adequate. It must also be borne in mind that increasing the set size increases

[22] This is not the only way to generate a code. The arrangement shown in figure 5.1 assigns higher multiples to the older measurements. The assignment of multiples can be reversed, and the recent measurements given more weight. In that case the code for 2121 would be 50. Is 2122 closer to 2121 than 1221 is? There is no definitive answer, but we assign more weight to slightly older measurements so that similar codes are more likely to share their common history.

[23] As a simple example, consider symbol sequences 1111 and 2000. They are neighbors in the code space but they are very dissimilar in the embedding space. Coding means converting the symbol sequence to a decimal code. Symbolization preserves the distance metric in an approximate way, but not exactly or mathematically.

the number of possible codes and sequences, thus rendering symbolization more computationally intensive. In many cases, after a certain parameter, the differential gain in contrast is not commensurate with the additional computations required, and a decision has to be made about this trade-off.

The researcher usually has to find the best symbolization parameters by the information about the system and some trial and error. Some methods about how to fine-tune the symbolization parameters can be found in Daw et al (1998).

The method that usually works involves computing certain quantities for various symbolization parameters in an ordered way and viewing the sharpness of the computed measure. An ordered way means that the user can change the set size while keeping the sequence length constant, or change the sequence length while keeping the set size constant. Usually the level of autocorrelation in the time series hints at the suitable symbolization interval. It is preferable to choose a higher symbolization delay for a time series with high first-order correlation.

6.4.1. Examples

This section shows a few examples about choosing the optimal symbolization parameters, using a bubble column time series for illustration. The starting choice for symbolization delay should be 1 unless it is known that the time series is highly oversampled and must be decimated. The information about the process and its memory informs us as to the sequence length. Since the bubble data can be reconstructed faithfully in three dimensions (albeit with higher delays[24]), and a good choice for the sequence length would be 3. There are not very clear guidelines about choosing the set

[24] In figure 4.15, the embedding delay used was 15.

size, but since the bubble column is a low-dimensional chaotic system, a set size of 2 or 3 may prove to be sufficient.

The effect of the symbolization parameters on the calculated Shannon entropy is now discussed. Their effect on mutual information is very similar[25]. In order to explore the time scales in the time series, it is suggested to compute the Shannon entropy for a range of symbolization intervals and for not just the symbolization interval of 1. Let $H_S(s,m,\tau)$ be the Shannon entropy for set size s, sequence length m and symbolization interval τ. For fixed symbol set size and sequence length, the plot of $H_S(s,m,\tau)$ versus τ allows one to graphically see the randomness or lack of predictability associated with the time-scale τ. Note that $H_S(s,m,1)$ is 1. Higher value of Shannon entropy signifies higher unpredictability, and it is convenient to plot the residual Shannon entropy, i.e., $H'_S(s,m,\tau) = 1 - H_S(s,m,\tau)$ so that a peak in the residual Shannon entropy signifies a timescale with high predictability. $H'_S(s,m,\tau)$ lies within [0, 1]. For example the residual Shannon entropy plot for a sine wave will have a peak for τ=T/4, T/2, 3T/4 and T where T is the period of the time series. Figure 6.2 shows the residual Shannon entropy plot for various symbolization parameters so that the sequence length is 3 and the set size ranges from 2 to 5. The time series used in figure 6.2 was collected on the bubble column and exhibited period-2 behavior; and was otherwise quite similar to that shown in figure 3.3 (a).

[25] Recall that mutual information is just the difference of Shannon entropies, and responds the same way to symbolization parameters as the former does.

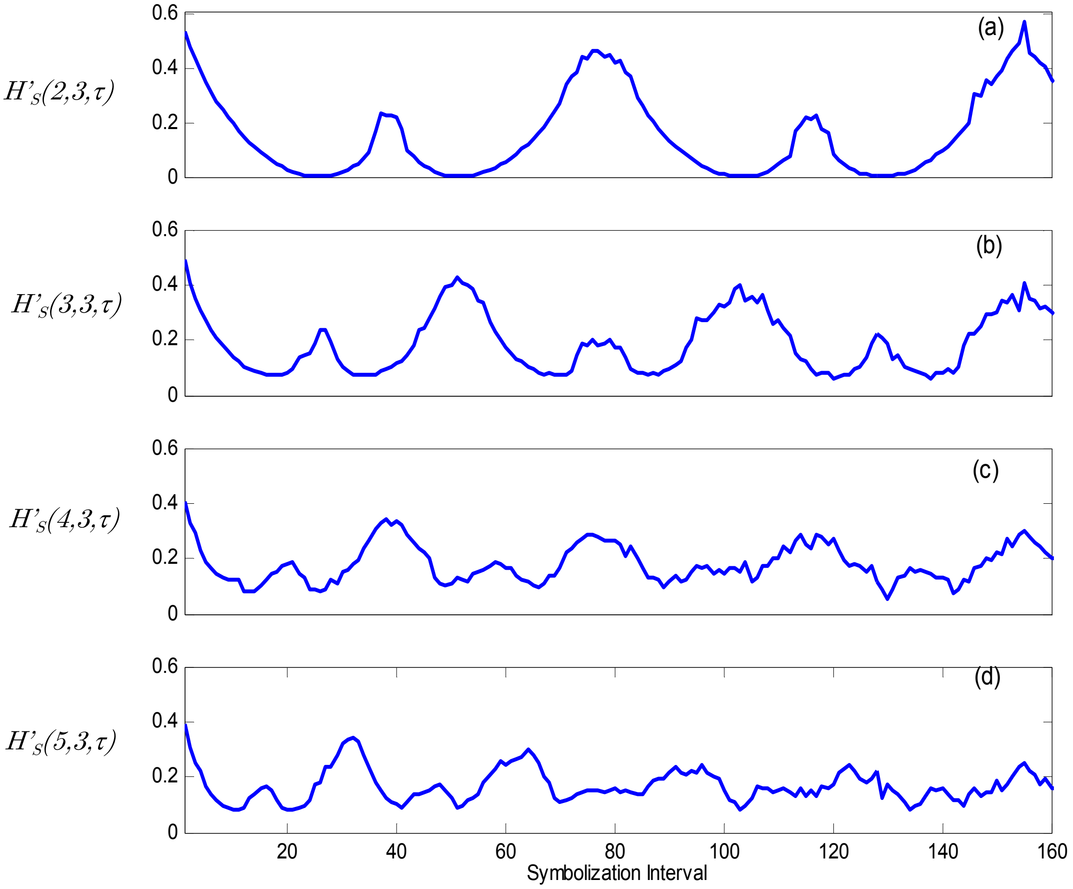


Figure 6.2. Residual Shannon entropy for various set sizes

The sequence length is 3 for all cases. The symbolization set size is 2, 3, 4 and 5 respectively for subplots (a), (b), (c) and (d). All subplots have the same scale. The time series appears in figure 3.3 (a). The series corresponded to gas flow rate of 170 cc/min and 0 V electrostatic potential.

It can be seen that the more peaks arise in the residual Shannon entropy plot as the set size is increased. Using a set size of 2 with a sequence length of 3 recognizes but eight patterns and it is no surprise that the resulting plot is very smooth. However, using a set size of 5 produces a rather flat curve, in which the salient features are difficult to distinguish. Also note that the difference between the minimum and maximum values taken by the residual entropy for set size of 2 is 0.6, which steadily declines as the set size is increased.

The reason that the residual entropy doesn't reach zero for set sizes greater than 2 is that the finer partition results in many possible codes, and the Shannon entropy is less likely[26] to be 1, and thus the residual entropy doesn't reach the value of zero. In other words, finer partition is introducing some sort of 'noise floor' in this case. Increasing the set size figure 6.2 (a) through (d) does not produce clearer patterns, or does not add information about the time series being studied.

The computational cost goes up exponentially upon increasing the set size, and that has to be borne in mind. The bubble column is a low-dimensional chaotic system, and the additional details seen by increasing the set sizes are probably small and unimportant. Based on figure 7.2 it can be concluded that given the sequence length of 3, the best set size is perhaps 2.

Figure 6.3 shows the entropy computed for the sequence lengths of 2 through 5 and the set size is fixed at 2 (the optimum set size just found). Increasing the sequence length produces more but sharper peaks. In figure 7.3 (a) through (d), there is no difference in the range of residual Shannon entropy values. However, the plots do appear sharp at the peaks and flat without for the sequence lengths of 4 and 5. A good choice for sequence length is perhaps 4 or 5 because for higher sequence lengths the gain in

[26] The Shannon entropy becomes 1 when only one bin is observed and with more codes becoming possible, the chances of that happening reduce.

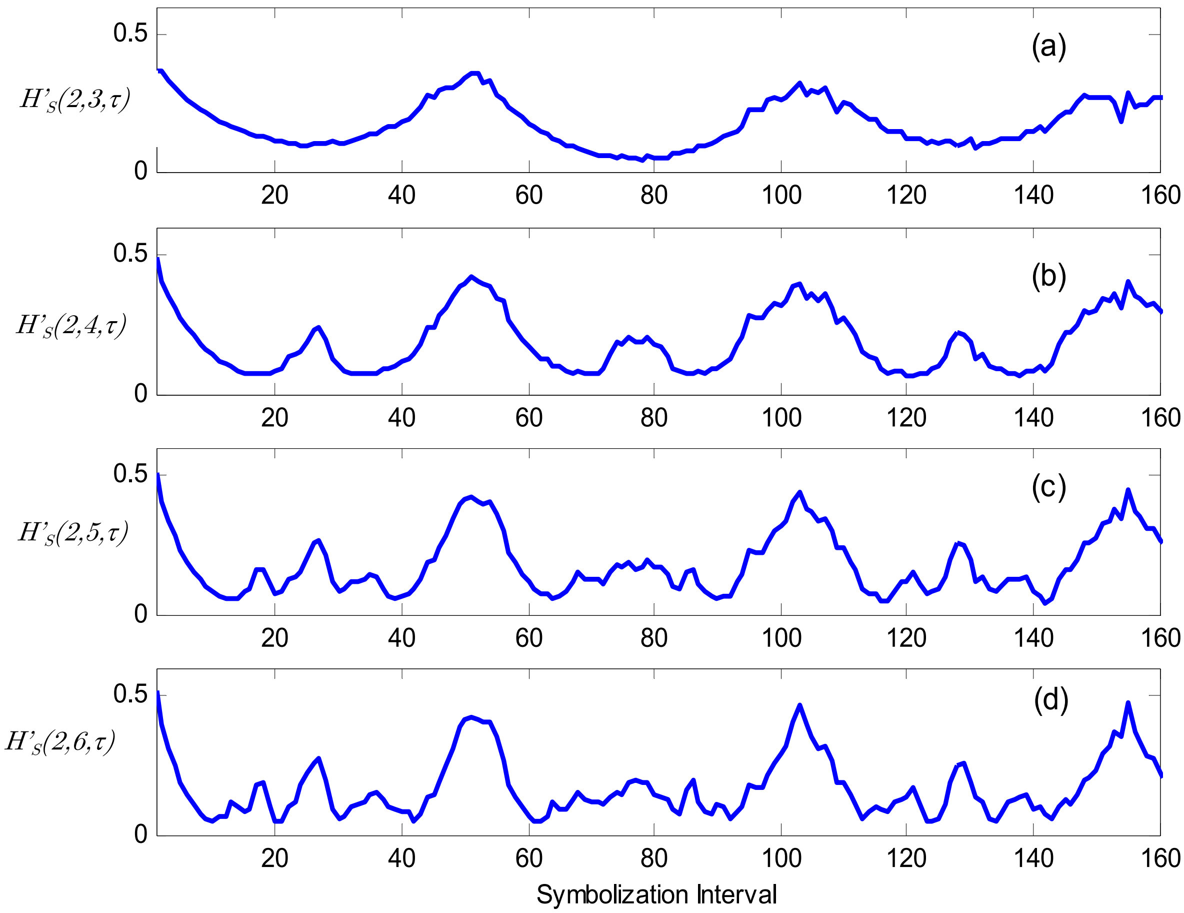


Figure 6.3. Residual Shannon entropy for various sequence lengths

The set size is 2 for all subplots. The sequence lengths for (a), (b), (c) and (d) are 3, 4, 5 and 6 respectively. The time series is shown in figure 3.3 (a).

contrast[27] is small but the computational cost mounts considerably.

Recall that the bubble column has a dimension between two and three. Although a three-dimensional embedding faithfully reproduces the geometry, Takens' limit is near 6 for the smallest embedding dimension that ensures faithful representation of the geometry. Information is obviously lost by symbolization, and hence the optimum sequence length is not 3 as one might expect because a three-dimensional embedding reproduces the geometry and eliminates self-crossings of the trajectories.

This chapter discussed entropy and mutual information, showed how they can be computed by way of symbolization, and provided a short review of the effect of symbolization parameters on the mutual information or Shannon entropy curves thus obtained. The next chapter considers the application of information theoretic measure for gauging stationarity and to compare different processes. The mutual information vector is treated as the raw feature vector for a pattern recognition problem. Note that the methodology outlined in the next chapter is applicable equally well to process monitoring and fault diagnosis. The relevant discussion can be found in section 5.6.

[27] By contrast, we mean that the plot is sharp at its peaks and is flat elsewhere

True genius resides in the capacity for evaluation of uncertain, hazardous and conflicting information

–Winston Churchill

Chapter 7

Applications of information-theoretic measures

The first minimum of mutual information is widely used to choose the embedding delay. It is also used to characterize the time series and gauge for the stationarity of the latter [Hively et al (2000)]. However, the first minimum of the mutual information contains the information about only the most dominant timescale. If several timescales are involved, merely the first minimum of mutual information may not be optimal. Chaotic time series for example often do not have a clear minimum and the measure cannot be defined for them. In contrast the computed mutual information function contains information about the general patterns, which makes it a useful feature vector for classification and identification.

Schreiber (1997B,1997A) attempted to analyze stationarity and to compare or classify time series using non-linear measures that depend on the similarity of one time series to another –and not just on the time series in question. It was remarked in Schreiber (1997B) that it would be useful to obtain a feature vector from a time series that adequately describes its dynamics. We propose that the mutual information function to be just that – a feature vector that uniquely and adequately characterizes a time series.

The goal of this chapter is to exploit the mutual information function to explore the stationarity of a time series and also to compare or classify different time series. The methods to reduce the dimension of the feature vector are introduced, followed by the description of an unsupervised learning technique called K-means algorithm. Finally, some examples are presented.

7.1. Reduction of dimensionality

Suppose a feature vector has dimension m. In most cases the number m is very large which leads to very high computational cost of pattern recognition, owing to the *curse of dimensionality*. Not all entries in the feature vector are useful for discrimination or characterization. Some entries may be very small, and some may be very similar for feature vectors pertaining to different classes. For all these reasons, it is a good idea to reduce the dimensionality of the raw feature vector before subjecting it to a classification algorithm.

There are many statistical methods for reducing the dimensionality of a multivariate vector. Some examples are PCA, Linear Discriminant Analysis (LDA), Canonical Correlation Analysis; and the standard (stepwise, forward and backward) variable selection methods. Of these, PCA is not optimal for class separation, and the variable selection methods assume certain distributional properties.

If the class of the feature vectors is known *à priori*, a simple information-based criterion may be used to choose the features. This method has the advantage over canonical analysis since it considers single dimensions and not whole set –and the reduced dimensionality in this fashion also eliminates some entries in the feature vectors, and obviates the need for computing them. The measure defined below, and used in this study, is taken from Watanabe and Kaminuma (1998) and called modified Fisher's information criterion.

If k indexes the feature number, and i and j refer to two different classes, then a Discriminant power of each feature $J_k(i,j)$ can be defined for every pair of classes as follows:

$$J_k(i,j) = J_k(j,i) = \frac{|| \mu_{i,k} - \mu_{j,k} ||^2}{{\sigma_{i,k}}^2 + {\sigma_{j,k}}^2} \text{ for } k = 1,2,..m \tag{31}$$

In the equation above, $\mu_{i,k}$ is the mean and $\sigma_{i,k}$ is the standard deviation of the k^{th} feature in the feature vectors corresponding to class i. The features with largest $J_k(i,j)$ are chosen as the features that best distinguish class i and j. If there are more than two classes, the final set of features will be the grand union of the subsets selected pairwise. However, this method cannot be used for unsupervised cases, but that does not hamper one from using it for classification.

7.2. K-means clustering algorithm

Clustering algorithms attempt to group similar points in a cluster based on some measure of similarity. The measure of similarity is usually taken as the Euclidean distance. Other measures of distance like Mahalanobis distance, infinite norm, block distances, etc. are also tenable. A reference for clustering algorithms is Fukunaga (1990).

K-means clustering algorithm minimizes the sum of squared distances of all data points in a cluster from their cluster center. The details for the K-means clustering algorithm are taken from Lou and Gonzalez (1974). A more comprehensive treatment of the algorithm is given in MacQueen (1967).

Step 1: Choose K initial cluster centers $\mathbf{z}_1(1)$, $\mathbf{z}_2(1)$,...,$\mathbf{z}_K(1)$.

Step 2: At the k^{th} iterative step distribute the samples $\{\mathbf{x}\}$ among the K cluster domains, using the relation

$$\mathbf{x} \in S_j(k) \text{ if } ||\mathbf{x}\text{-}\mathbf{z}_j(k)||<||\mathbf{x}\text{-}\mathbf{z}_i(k)|| \text{ for all } i=1,\ 2,\ ...,\ K$$

Where $S_j(k)$ denotes the set of samples whose cluster center is $\mathbf{z}_j(k)$. Ties may be resolved arbitrarily.

Step 3: From the results of step 2, compute the new cluster centers $\mathbf{z}_j(k+1)$, j=1, 2, ..., K, such that the sum of the squared distances from all points in $S_j(k)$ to the new cluster center is minimized. In other words, the new cluster center $\mathbf{z}_j(k+1)$ is computed so that the performance index $J_j = \sum_{\mathbf{x}\in S_j(k)} ||\mathbf{x}\text{-}\mathbf{z}_j(k+1)||^2,\ j=1,\ 2,\ ...,K$ is minimized. The value of $\mathbf{z}_j(k+1)$ which minimizes this performance index is simply the sample mean of $S_j(k)$. Therefore the new cluster center is given by

$$\mathbf{z}_j(k+1) = \frac{1}{N_j}\sum_{\mathbf{x}\in S_j(k)} \mathbf{x}\ ,\ j = 1,\ 2,\ ...,K$$

where N_j is the cardinality of the set $S_j(k)$. The name K-means is obviously derived from the manner in which cluster centers are sequentially undated.

Step 4: If $z_j(k+1)=z_j(k)$ for $j=1, 2, ..., K$, the algorithm has converged and the procedure is terminated. Otherwise go back to Step 2.

The behavior of the K-means algorithm is influenced by the number of cluster centers specified, the choice of initial cluster centers, the order in which the samples are presented, and, of course, the geometrical properties of the data. Although no general proof of convergence exists for this algorithm, it can be expected to yield acceptable results when the data exhibit characteristic pockets which are relatively far from each other. In most practical cases the application of this algorithm will require experimenting with various values of K as well as different choices of starting configuration. A good practice is to run the algorithm several times with the same K, and presenting the feature vectors to the algorithm in different, randomized order. If the algorithm converges to very similar results, then it can be assumed that the clustering is robust and not very sensitive.

In the K-means algorithm the probability that a point belongs to a certain cluster (or the membership of a point to any cluster) is binary, -a point either is in a cluster or not. However, the K-means algorithm can be *fuzzified*, i.e., any data point can have membership to any of the cluster centers in such a way that the sum of its memberships to all clusters is unity[28]. That algorithm, known as Fuzzy C-means clustering, is also quite popular. If desired, one could apply a hard cut-off in the final step of Fuzzy C-means clustering (winner-take-all strategy) so that the results of clustering algorithm are crisp and not soft or fuzzy.

The advantage of using fuzzy clustering is that if there are errors in the algorithm, one can observe the membership of the feature vector wrongly

[28] It is not necessary because fuzzy memberships are not probabilities. The practice though is common perhaps because it makes intuitive sense.

classified to various clusters to extract some information about the similarity of that vector to various clusters. Assume that a wrongly classified vector **q** came from class A but was assigned to cluster B, which is the cluster representing class B. If q had 49% membership to the cluster A, but the remaining 51% of the membership was to cluster B, it shows that there is a healthy amount of confusion in the clustering algorithm about the membership of **q**. If desired, one can then go back and scrutinize the time series segment that gave rise to **q**. On the other hand if **q** has only 2% of the membership to A, it means that something is seriously wrong with **q**. It certainly is not a representative vector of its class in the reduced dimension, and the time series segment that produced it must be re-examined.

Fuzzy clustering is not used in this dissertation due to proverbial space limitation. At the time of writing, more research is underway on this subject.

7.3. Gauging stationarity

In this section, we discuss how the mutual information function can be used to test for stationarity, and show some examples. First, the mutual information function is computed for the segments formed from the first and last one-third of the series. The mutual information function will be the raw feature vector, and the feature vectors pertaining to the first one-third of the data constitute class A and those corresponding to the last one-third of the data constitute class B. Fisher's information criterion as defined in equation (36) is then used to reduce the dimensionality. Note that the reduced dimensional feature vector contains the mutual information function at certain lags, which makes this approach suitable for online application since one needs to calculate mutual information for only certain values of lag, which can be easily done.

The reduced feature vectors are then subjected to K-means clustering algorithm with the stipulation to find two clusters. Later an estimate of error is obtained in order to quantify the success of the algorithm. The classification error is the fraction of objects wrongly classified. If most of the objects in a cluster are from class A, it is assumed that the cluster is contains feature vectors or objects from class A.

If the resulting error is 0%, the time series is clearly non-stationary. On the other hand, if the error is 50%, then the time series is very stationary because a feature vector from any part of the time series is equally likely to be in class A or class B. This method obviously does not lend itself to confirmatory analysis, but the classification error will give us a very good estimate of the extent of non-stationarity in the time series. The smaller the error rate, the more likely a time series is to be non-stationary.

To obtain unbiased estimates of error, cross-validation techniques can be used, or the data can be partitioned into training and testing data (hold-off). Whether one can afford the luxury of partitioning the data depends on the amount of data available relative to the time scales in the time series. The classification error reported in this chapter is unbiased, because a fraction of the feature vectors is held off for validation. Roughly 60% to 70% of the feature vectors were used for training or clustering, and the remaining 30% to 40% constituted the hold out sample, on which the classification error was calculated.

For the first illustration a bubble column time series is used. A segment of the time series appears in figure 3.3 (b). Based on the results in section 5.2, the time series is likely non-stationary. For illustration, figure 7.1 shows two segments each from the first and last one-thirds of the time series.

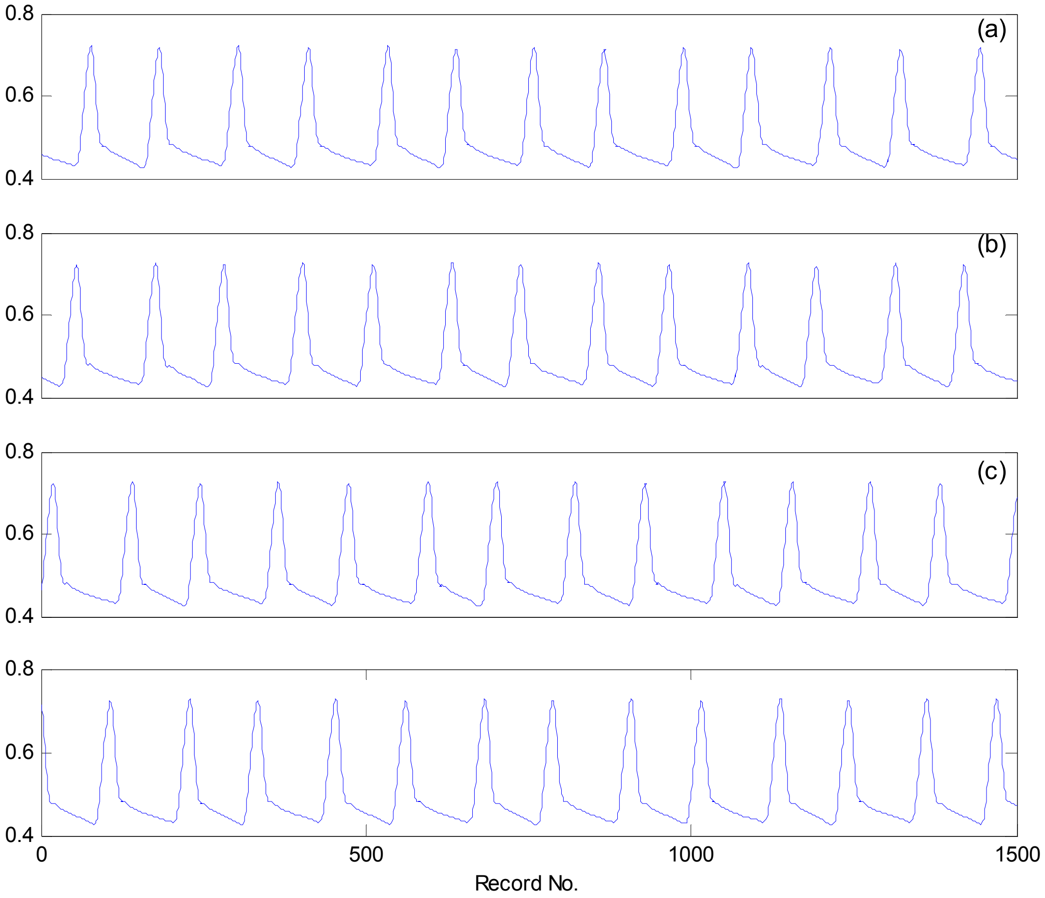


Figure 7.1. Four segments from a bubble column time series

Segments (a) and (b) are from the first one-third and segments (c) and (d) are taken from the last one-third of the series. The flow rate for this data set was 170 cc/min and the corresponding electrostatic potential was 12 kV. The abscissas are the differential pressure measurements across the nozzle.

The difference between the first and last one-thirds is not very apparent. Recall that figure 5.2 showed the return maps for the first and last one-thirds of this time series. The return maps occupied roughly the same zone, but there were structural differences between the first and last one-thirds.

Mutual information was computed for seven segments each of the time series formed from the first and last one-thirds. Symbol set size of 2 and the sequence length of 5 was used[29]. Symbolization interval was taken to be 1. The mutual information function plots are presented in figure 7.2.

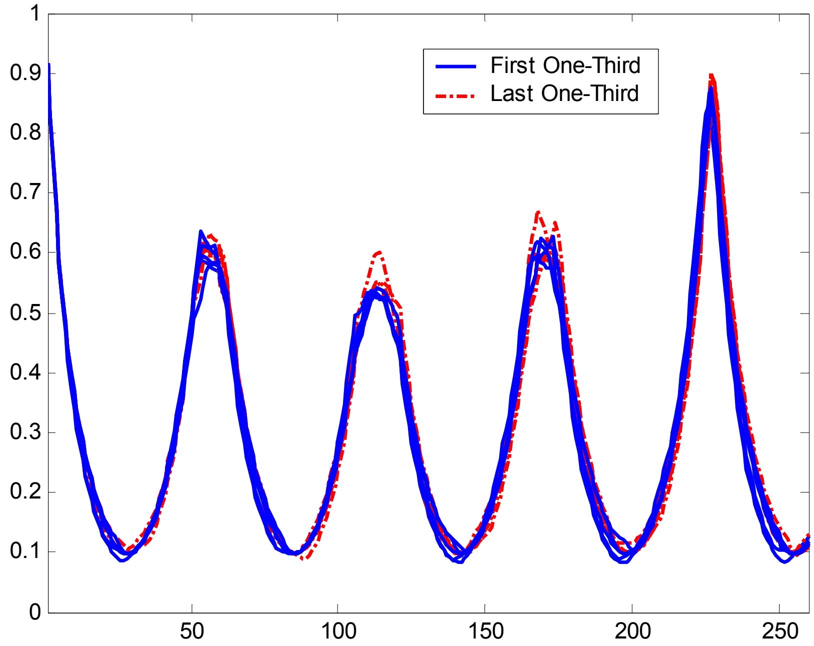


Figure 7.2. Mutual information plots for the time series in figure 7.1

The dashed plots are for the last one-thirds of the series, and the solid plots are for the first one-thirds.

[29] These parameters were found to be optimum in section 6.4

Modified Fisher's Information criterion was used to find the most significant features[30]. Figure 7.3 illustrates the computed Fisher's information criteria for the first and last one-thirds of the time series. The first two dimensions of the reduced dimensional feature vectors are shown in figure 7.4. Only five vectors for each class that were submitted to the clustering algorithm are shown in figure 7.4.

Note that all but one of the filled squares corresponding to the last one-third of the time series are in the top right of figure 7.4, and it seems that a straight line can correctly classify all feature vectors even in two dimensions. With only five features, the two segments were correctly

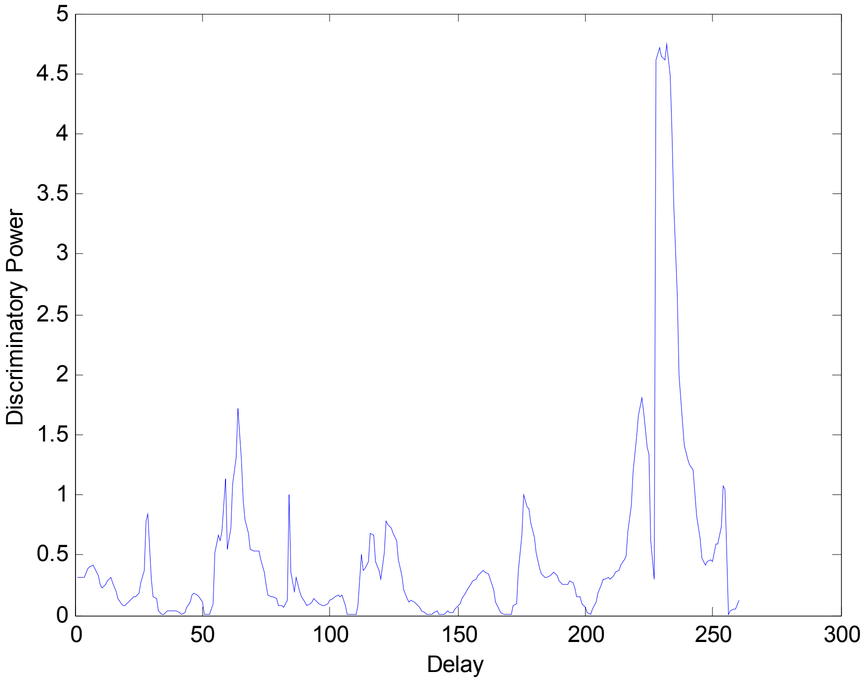


Figure 7.3. Discriminatory power for the feature vectors in figure 7.2

[30] In this chapter, 'feature' means the mutual information function for a certain delays (τ–values). Since every entry in the raw feature vector is the mutual information at a certain delay, the reduced-dimensional feature vector contains only the mutual information at certain delays.

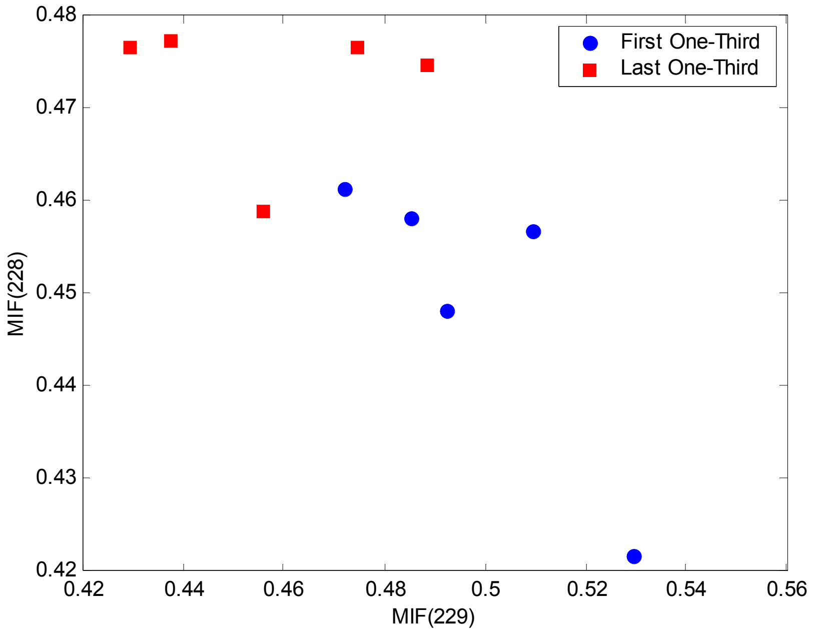


Figure 7.4. Reduced dimensional feature vectors (bubble column)

The ordinate and the abscissa are mutual information at τ=228 and 229 respectively

classified. Two feature vectors (out of 7) for each class were kept for validation. The K-means algorithm was run 10 times to average out the classification error (on the hold-out sample). With only 5 features selected, the classification error was 0%. Thus, if with only 5 feature vectors the clustering algorithm classified the feature vectors from the first and last one-thirds of the data set without error, one should have strong suspicions about the stationarity of the time series. It was known however that this time series was not stationary, and the return map in figure 5.1 demonstrated that.

The next illustration uses data from a laboratory fluidized bed. A brief description of the fluidized bed setup is provided in section 3.2. The time series used for illustration was known to be stationary, since no changes were made in the operating conditions while the data was collected. The time series is shown in figure 3.6 (b). Sequence length of 5 and set size of 3 was used for symbolization. Symbolization interval used was 1. Nine feature vectors (or segments) were formed from each of the first and last one-thirds of the series. Of the 18 feature vectors available, 12 were used for training and 6 were used for validation. Modified Fisher's Information criterion was used to select the features from the raw mutual information vectors. Figure 7.5 shows first two dimensions of the reduced feature vectors. Using only 10 features resulted in 41.67% error rate[31].

Increasing the number of features selected to 50 resulted in the same error rate. Based on that, one can be reasonably confident in saying that the

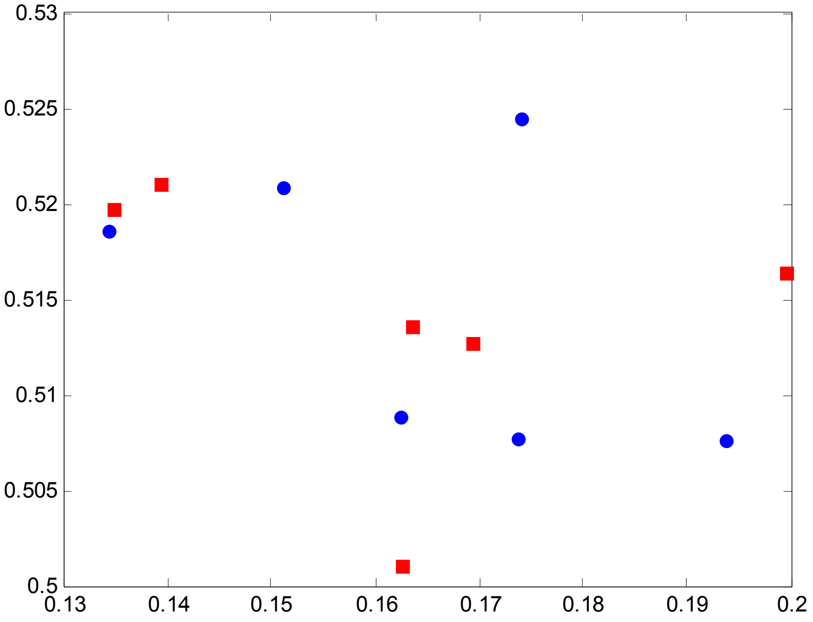


Figure 7.5. Reduced dimensional feature vectors (fluidized bed)

The circles and squares correspond to the feature vectors formed from the first and last one-thirds of the time series in figure 3.6(b). See text for detail.

[31] Half of the runs produced 3 errors out of 6, and half produced 2 errors out 6. The average classification error is thus 41.67%.

time series is stationary. At any rate, the series is not non-stationary. It is clear that the feature vectors have considerable amount of overlap. Apparently this overlapping did not lessen in higher dimensions –thus the higher error of classification.

In this section it was shown how the mutual information function can be used to gauge the stationarity of a time series. The first example was that of a non-stationary time series, and the non-stationarity was detected. The second example was of a time series that was deemed stationary, or not non-stationary. The approach has been useful, but one still has to find the optimum parameters for symbolization.

7.4. Comparing different processes

The approach outlined in the previous section to gauge stationarity can be used to compare two different processes. It must however be assumed that the processes being compared are stationary. The feature vectors from one process can be assigned to class A, and those from the other process can be classified as class B. This method is not limited to two processes. Any number of different processes can be compared using the K-means algorithm.

Figure 7.6 shows three time series from a fluidized bed. The gas flow rates corresponding to these time series are 1.49, 1.65 and 1.85 cc/s. The two nearest states differ in gas flow rate by roughly 11%. All the examples in this section were collected on a fluidized bed. A brief description of the experimental setup is provided in section 3.2.

To compute the mutual information, symbol set size of 3, sequence length of 5, and symbolization interval of 1 was used. The dimension of these raw feature vectors was reduced according to the modified Fisher's information criterion. Figure 7.7 shows the resulting feature vectors plotted

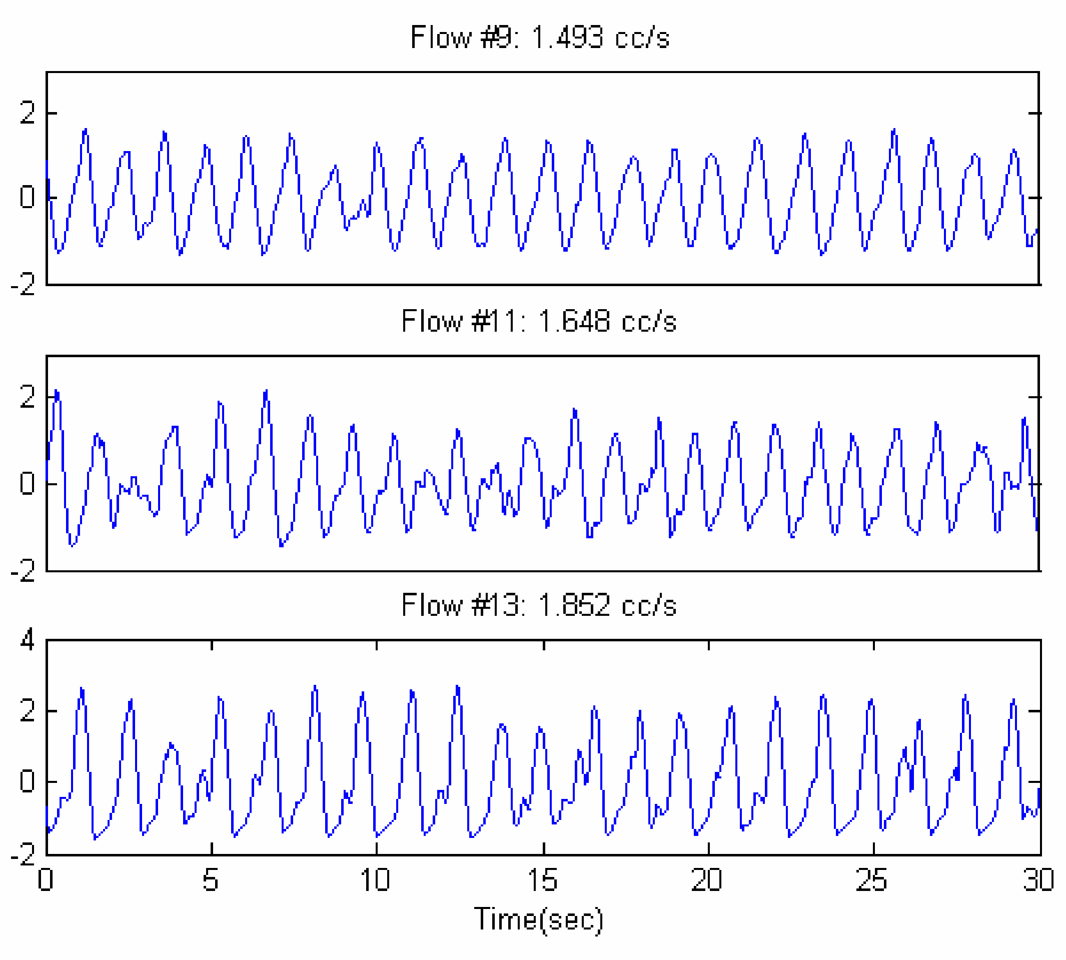


Figure 7.6. Three fluidized bed time series (9, 11 and 13)

The abscissas contain the differential pressure as described in section 3.2.

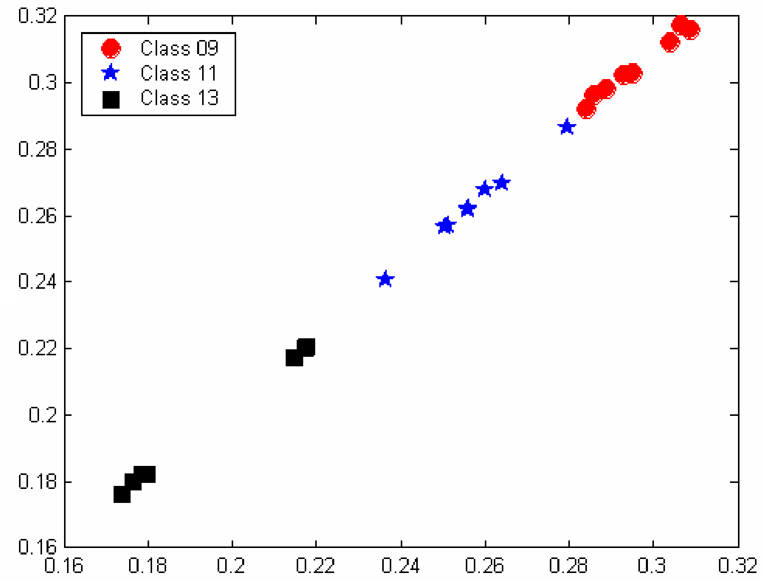


Figure 7.7. Three dynamic states (9, 11 and 13) in reduced state space

The abscissas contain the differential pressure as described in section 3.2.

in the first two dimensions. The three classes appear easily separable in only two dimensions. Even with 3 features selected, the clustering algorithm classified the holdout sample without error.

It shows us that the mutual information function can classify and characterize the time series that are not very conspicuously different in their spectral densities. Figure 7.8 shows spectral densities for the series in figure 7.6. The spectral densities were computed based on 8192-point FFT windowed with a 4096-point Hanning window. The density was computed on 50,000 data points sampled at 100 Hz. First and last 5,000 data points in each time series were discarded to avoid any possible transients.

The densities in 7.8 (a) and 7.8 (b) do not seem to be very different. A shift in power towards lower frequencies is visible in figure 7.8 (c). However, note that the confidence intervals are very wide. When the spectral density

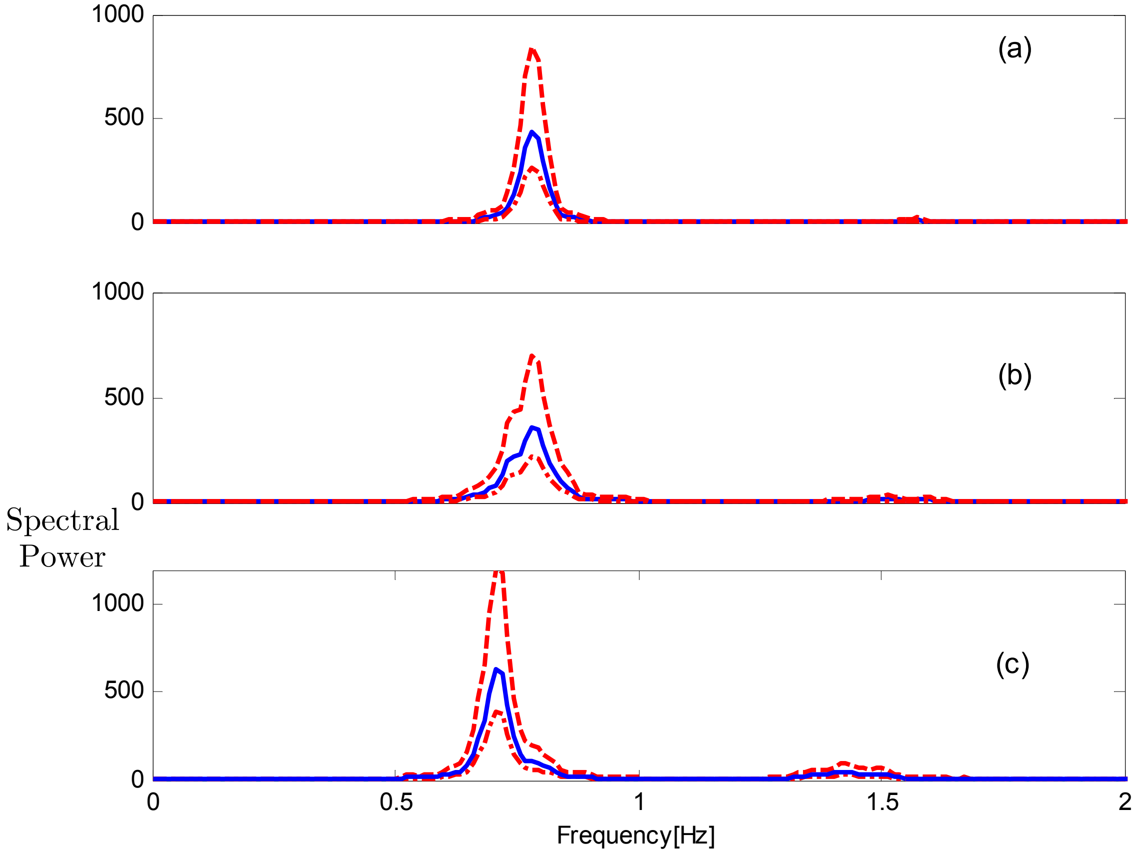


Figure 7.8. Spectral densities for the series in figure 7.6

Subplots (a), (b) and (c) correspond to the gas flow rates of 1.49, 1.65 and 1.85 cc/s, respectively. The solid line is the average power, and the dashed lines are the 95% confidence intervals.

vectors were subjected to the K-means algorithm for classification, the classification error was 25% when power at 20 frequencies (chosen according to Fisher's modified information criterion) was used as the feature vector. Increasing the number of features didn't reduce the classification error. This example demonstrates that the mutual information function is capable of distinguishing time series that can not be adequately characterized with power spectral density.

The next example attempts to compare three classes that differ in the gas flow rate by only 6%. The corresponding flow rates were 1.49, 1.54 and 1.65 cc/s. The time series are shown in figure 7.9. Same symbolization parameters as the previous example were employed. Figure 7.10 contains the reduced dimensional feature vectors in the first two dimensions.

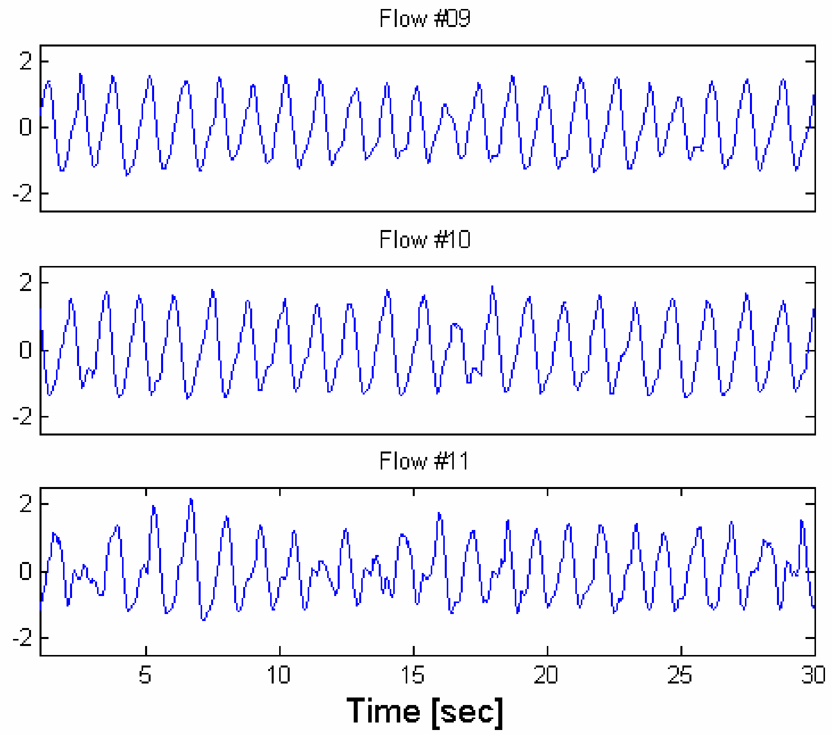


Figure 7.9. Three fluidized bed time series (9, 10 and 11)

The abscissas contain the differential pressure as described in section 3.2.

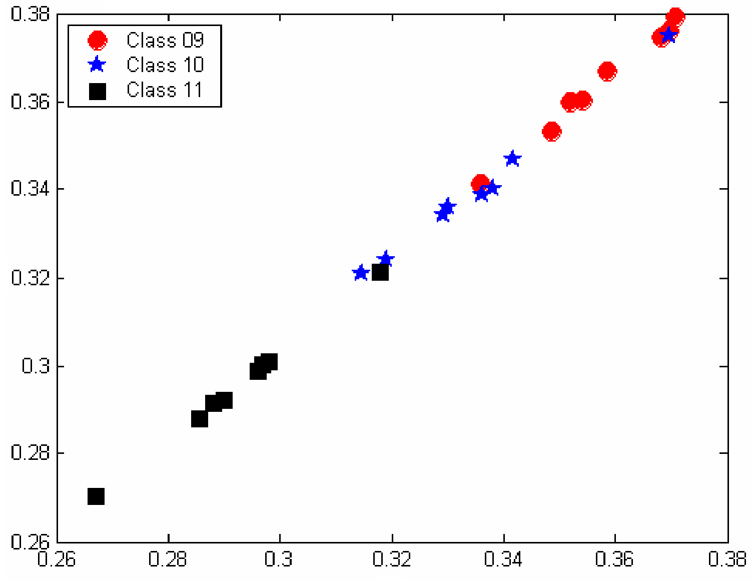


Figure 7.10. Reduced dimensional feature vectors (9, 10 and 11)

The time series are not very different, and indeed there power spectral densities were very similar. Spectral densities for two of these time series (9 and 11) are shown in figure 7.8 (a) and (b) respectively. The classification error when 15 features were used was 9%. Increasing the features did not reduce the classification error, and if too many features were selected, the performance of the algorithm actually deteriorated. However, a classification error of 9% is much better than that of 45% obtained using the spectral density vectors. When we tried to distinguish the classes pair wise, the classification error reduced to zero with 15 features.

We tested the limits of our algorithm by attempting to compare three very similar chaotic states in a fluidized bed at higher velocity. One must recall that the mutual information function for a chaotic time series usually reaches a plateau for small lags and there are not many features in the mutual information function of chaotic time series to differentiate two very similar chaotic states. The time series are shown in figure 7.11.

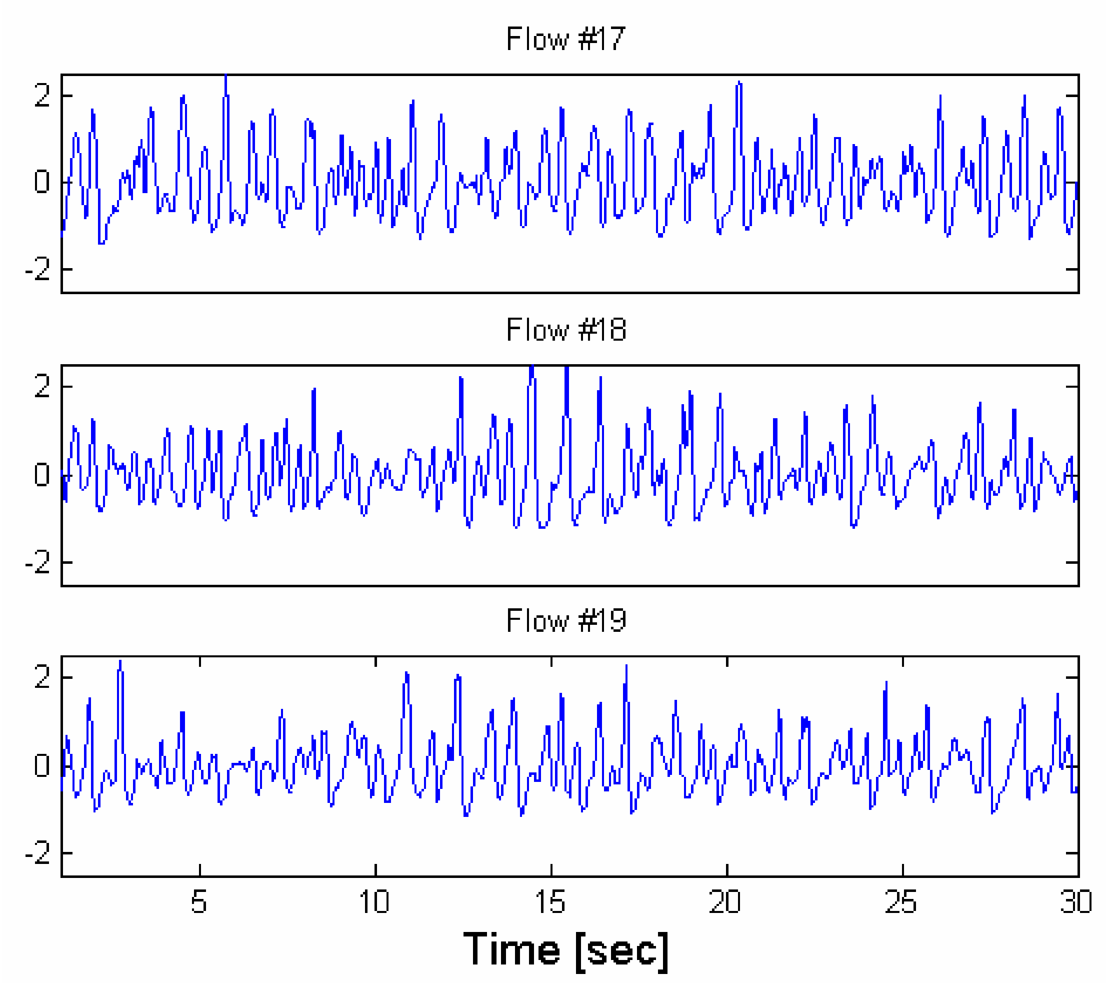


Figure 7.11. Three high-velocity fluidized bed time series (17, 18 and 19)

The abscissas contain the differential pressure as described in section 3.2.

The flow rates corresponding to these time series are 2.58, 2.84 and 3.09 cc/s respectively. The feature vectors are shown in figure 7.12.

It seems that there is very little difference between these time series. The classification algorithm did not correctly classify all three classes. The classification error when 15 features were used was 29%.

This chapter showed how the mutual information function can be used to gauge the stationarity of a time series and to compare different processes. The extension to process monitoring is straightforward, and the relevant comments can be found in section 5.4. It was seen that the mutual information function could successfully distinguish even similar flow rates,

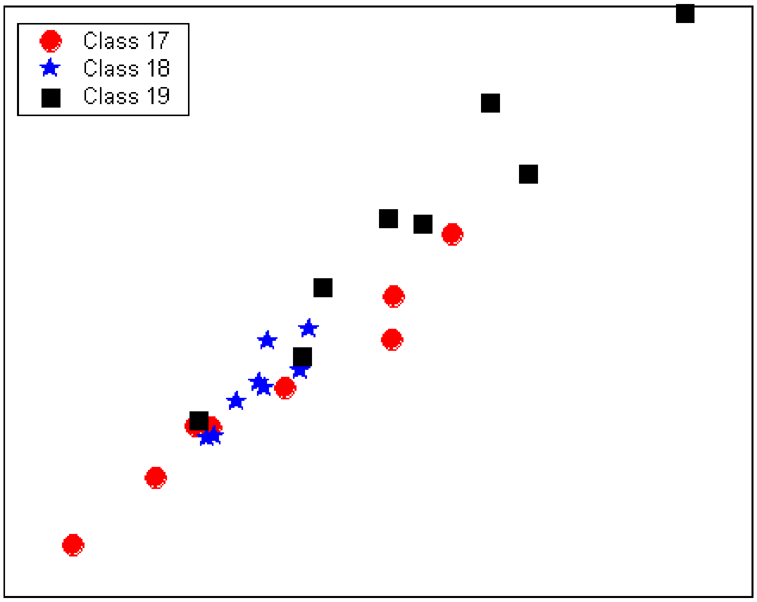


Figure 7.12. Reduced dimensional feature vectors (17, 18 and 19)

The time series segments are shown in figure 7.11

but it was not very powerful when used to compare very similar high-velocity states. There is reason to believe that perhaps other clustering algorithms will perform better. The Max-Min algorithm was also used on the reduced feature vectors (the feature vectors that were input to the K-means algorithm), but it provided poor clustering performance. Further research on this subject is underway.

With *à priori* information about the process, a library of system states or of known faults can be created. With the dimension-reduction technique used, the features retained are essentially the mutual information at some delays that can be quickly computed online. The feature vector thus formed from a time-window covering recent observations can be compared with the library of known faults or of system states. This allows one to detect if the system is moving towards a known fault and also to optimize maintenance costs. A measure of dissimilarity between the current feature and the cluster describing a desirable system state can be used for identification. The ability of the mutual information function to characterize the system and to detect changes has important implications for control. The comments in section 5.6 are applicable to the feature vectors formed by mutual information.

The next and final chapter briefly mentions the original work in this study and proposes relevant future directions.

One must be a god to be able to tell successes
from failures without making a mistake.
-Anton Chekhov

‘ *“The time has come,” the Walrus said,*
“To talk of many things: ... ’
-Lewis Carroll

Chapter 8

Conclusions and future directions

8.1. Conclusions

This dissertation makes three major original contributions.

1. It introduces and develops cluster-linked principal curve (CLPC) algorithm, which is computationally much less expensive than Hastie and Stuetzle's Principal curve (HSPC) or other similar algorithms. The CLPC algorithm is truer in spirit to the Expectation-Maximization (E-M) principle, because it treats all the dimensions together (without specifying dependent and independent variables) and not separately or pairwise like other Principal Curve algorithms do.

2. It demonstrates that the distributions of arc lengths along the CLPC characterize a time series based on their return maps and delay embeddings, by testing for stationarity and reversibility. It also outlines how this framework can be applied to monitoring and forecasting.

3. It utilizes the mutual information function instead of the commonly used ‘first minimum’ of mutual information to characterize a time series. The discussion about choosing the best symbolization parameters provides an improvement over the popular methods for the same purpose by considering the information theoretic quantities for various symbolization intervals.

8.2. Practical applications

The techniques developed in this dissertation are very suitable for process monitoring and fault diagnosis problems, as was discussed in section 5.4. The CLPC framework can also be extended to cover forecasting (cf. section 5.5). It was demonstrated how these methods can be used to detect changes in global dynamics, which is tantamount to process monitoring. It was also argued that with a given library of system states and/or known faults, the methods can perform, respectively, system identification and fault diagnosis. The ability to accurately detect change in dynamics definitely has important implications for control.

The techniques are suitable for online applications as well, since the required online computations are reasonable. For example, calculating the mutual information for some delays or projecting the data points on a polygonal line is neither time-consuming nor iterative.

These methods were motivated by examples taken from nonlinear systems, but can be applied to linear systems as well. Even in presence of small amounts of nonlinearity, fault diagnosis and system identification systems that draw on the techniques introduced in this dissertation will achieve superior performance as compared to that possible with linear methods alone.

8.3. Future directions

For the future, it would be useful to more rigorously study the mathematics behind the CLPC algorithm, and to extend it so that it may capture the fractal structure of strange attractors better than the current algorithm. Extending the CLPC framework for prediction appears to hold promise as well. It may also be worthwhile to try out other clustering algorithms in attempts to compare time series based on their mutual information vector, or other information-theoretic measures characterizing the time series such as Kullback-Leibler Information. It is suggested that characterization requiring specification of some parameters (e.g. symbolization parameters), be formulated as an iterative optimization problem.

Bibliography

Atkeson, C.G., Moore, A.W. and Schaal, S. (1997), Locally weighted learning, *Artificial Intelligence Review* **11** 1, pp. 11-73.

Box, G. E. P., and Jenkins, G. (1976*), Time Series Analysis: Forecasting and Contro*l, Holden-Day, San Francisco, CA.

Cleveland, W.S. (1979), Robust locally weighted regression and smoothing scatterplots, *J. Am. Stat. Assoc.*, **74,** 829-836.

Crutchfield, J. P. and Packard, N. H. (1983), Symbolic dynamics of noisy chaos, *Physica D*, **7,** 201.

Daw, C. S., Finney, C. E. A. and Tracy E. R. (2001), Symbolic analysis of experimental data, *Review of Scientific Instruments*.

Daw, C. S., Kennel, M. B., Finney, C. E. A. and Connolly, F. T. (1998), Observing and modeling nonlinear dynamics in an internal combustion engine, *Physical Review E,* **57**, 2811-2819.

Daw, C. S., Finney, C. E. A., Vasudevan, M., van Goor, N. A., Nguyen, K., Bruns, D. D., Kostelich, E. J., Grebogi, C., Ott, E. and Yorke, J. A., Self-organization and chaos in a fluidized bed, *Physical Review Letters*, **75**(12), 2308-2311.

Delicado, P. (2001), Another look at principal curves and surfaces, *Journal of Multivariate Analysis*, **77**, 84-116.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977), Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Stat. Soc. B*, **39**(1), 1-38.

Diks, C. (1999), *Nonlinear Time Series: Methods and Applications*, World Scientific, New York, NY.

Dong, D. and McAvoy, T. J. (1996), Nonlinear principal component analysis: based on principal curves and neural networks, *Comp. Chem. Engg.*, **20** (1), 65-78.

Duchamp, T. and Stuetzle, W. (1996), Extremal properties of principal curves in the plane, *Annals of Statistics*, **24**(4), 1511-1520.

Duchamp, T. and Stuetzle, W. (1996), Geometrical properties of principal curves in the plane, in Helmut Rieder (ed.), *Robust statistics, data analysis, and computer intensive methods: in honor of Peter Huber's 60th birthday*, v. 109 of *Lecture notes in Statistics*, pp. 135-152, Springer-Verlag, New York, NY.

Fraser, A. M.(1989), Information and entropy in strange attractors, *IEEE Trans. Inf. Theory*, **35**(2), 245-262.

Fraser, A. M. and Swinney, H. L., (1986), Independent coordinates for strange attractors from mutual information, *Phys. Rev. A*, **33**(2), 1134-1140.

Fukunaga, K. (1990), *Introduction to Statistical Pattern Recognition*, Academic Press, New York, NY.

Gnanadesikan, R. (1964), *Methods for statistical data analysis of multivariate measurements*, John Wiley, New York, NY.

Grassberger, P., Schreiber, T., and Schaffarath, C. (1991), Nonlinear time series analysis, *Int. J. Bifurc. Chaos*, **1**, 521.

Grassberger, P. and Poccacia, I. (1983), Measuring the strangeness of strange attractors, *Physica D*, **50**, 189.

Hastie, T., and Stuetzle, W., (1989), Principal curves, *J. Am. Stat. Soc.*, **84**(406) , 502-516.

Hively, L. M., Protopopescu, V. A. And Gailey, P.C. (2000), Timely detection of dynamical changes in scalp EEG singals, *Chaos*, **10**(4), 864-875.

Isliker, H., and Kurths, J. (1993), Nonlinearities and nonstationarities in stock returns, *Int. J. Bifurc. Chaos*, **3**, 1573.

Kantz, H., and Schreiber, T. (1997), *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge, U.K.

Kégl, B. (1999), "Principal Curves: Design, learning and applications," Ph.D. Dissertation, Concordia University, Montréal, Québec, Canada.

Kennel, M., and Mees, A. I. (1998), "Testing for general dynamical stationarity with a symbolic data compression technique," *Technical report, INLS, UC San Diego*, November 1998.

Kennel, M. (1997), Statistical test for dynamic non-stationarity in observed time series data, *Phys. Rev. E.,* **56**, 1.

Kennel M., Brown, R., and Abarbanel, H. D. I. (1992), Determining embedding dimension for phase-space reconstruction using a geometrical construction, *Phys. Rev. A,* **45**, 3043.

Kostelich, E., and Yorke, J. A. (1988), Noise reduction in dynamical systems, *Phys. Rev. A,* **38**, 1649.

Lou, J. T., and Gonzalez, R. C. (1974), *Pattern Recognition Principles*, Addison-Wesley, New York, NY.

MacQueen, J. (1967), Some methods for classification and analysis of multivariate data, *Proceedings of the 5th Berkeley Symposium on Probability and Statistics*, University of California Press, Berkeley, CA.

Manuca, R., and Savit, R. (1996), Stationarity and non-stationarity in time series analysis*, Physica D*, **99**, 134-161.

Menako, C. R. (2001), "An experimental study of linear and chaotic bubble formation in liquid-filled columns under electrostatic potentials," M. S. Thesis, The University of Tennessee.

Needham, J. (1959), *Science and Civilisation in China*, Vol. III, Cambridge University Press, Cambridge, U.K.

Paluš, M. (1995), Testing for non-stationarity using redundancies: qualitative and quantitative aspects, *Physica D*, **80**, 186.

Pawelzik, K. and Schuster, H. G. (1987), Generalized dimensions and entropies from a measured time Series, *Phys. Rev. A.*, **35**, 481.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T.(1993), *Numerical recipes in C: the art of scientific computing*, Cambridge University Press, Cambridge, U.K.

Prichard, D., and Theiler, J. (1995), Generalized redundancies for time series analysis, *Physica D*, **84**, 476.

Priestley, M. B. (1989), *Nonlinear and Nonstationary Time Series Analysis*, Academic Press, London, U.K.

Rajput, S. And Bruns, D. D. (2003), Principal curves and chaos, *Proc. 7th Exptl. Chaos Conf.*, San Diego, CA, Aug 25-29, 2002.

Rajput, S., Bruns, D. D., Menako, C. R. and DePaoli, D. J. (2002), "Chaotic dynamics of bubble formation from electrified capillaries," *AIChE Annual Meeting 2002*, Indianapolis, IN, Nov. 4-9, 2002.

Rajput, S., and Bruns, D. D. (2001B), "Nonlinear measure based process monitoring and fault diagnosis," *AIChE Annual Meeting*, Reno, NV, November 4-9, 2001.

Rajput, S., and Bruns, D. D. (2001A), "Quantifying nonstationarity in a nonlinear bubble column," *AIChE Annual Meeting*, Reno, NV, November 4-9, 2001.

Rössler, O. E. (1976), An equation for continuous chaos, *Physics Letters A* **57**, 397-398.

Savit, R. and Green, M. (1991), Time series and dependent variables, *Physica D*, **50**, 95.

Schreiber, T. (2000), Measuring information transfer, *Phys. Rev. Lett.*, **85**, 417.

Schreiber, T. (1997), Classification of time series data with nonlinear similarity measures, *Phys. Rev. Lett.*, **79**, 1475.

Schreiber, T. (1997), Detecting and analysing nonstationarity in a time series with non-linear cross-predictions, *Phys. Rev. Lett.,* **78**, 843.

Schreiber, T. and Schmitz, A. (1996), Improved surrogate data for nonlinearity tests, *Phys. Rev. Lett.,* **77**, 635.

Shannon, C. E. (1948), A mathematical theory of communication, *The Bell System Technical Journal*, **27**, 379-423, 623-656.

Silverman, B.W. (1985), Some aspects of spline smoothing approaches to non-parametric regression curve fitting, *J. Royal. Stat. Soc. B*, **47**, 1-52.

Takens, F., *in* Rand, D., and Young, L. S. (eds) (1981), *Dynamical Systems and Turbulence* in *Lecture notes in Mathematics 898*, pp 366, Springer, Berlin.

Tang, X. Z. and Tracy, E. R. (1998), Data compression and information retrieval via symbolization, *Chaos*, 8(3), 688-696.

Theiler, J., Eubank, S., Longtin, A., Gadrikian, B., and Farmer, J. D. (1992), Testing for non-stationarity in time series: the method of surrogate data, *Physica D*, 58, 77.

Theiler, J. (1986), Spurious dimensions from correlation algorithms applied to limited time series data, *Phys. Rev. A*, **34**, 2427.

Tibshirani, R. (1992), Principal curves revisited, *Statistics and Computation*, **2**, 183-190.

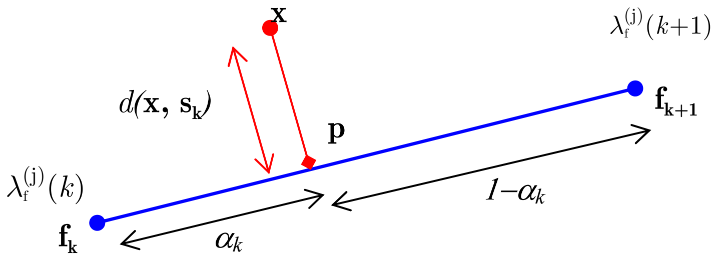
Tong, H. (1990), *Nonlinear Time Series: A Dynamical Systems Approach*, Clarendon Press, Oxford.

Watanabe, S., and Kaminuma, T. (1998), Recent developments of the minimum entropy algorithms, *Proceedings of the International conference on Pattern Recognition, IEEE*, pp 536-540.

Appendixes

A. Projecting a point on the Principal Curve

Let **x** be the data point to be projected on the polygonal line formed by connecting the cluster centers $\{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_{nc}\}$. The line segment formed by joining $\mathbf{f}_k$ and $\mathbf{f}_{k+1}$ is referred to as $\mathbf{s}_k$. Consider the case when it is desired to find the projection of **x** on the line segment formed by joining $\mathbf{f}_k$ and $\mathbf{f}_{k+1}$. The cluster centers $\mathbf{f}_k$ and $\mathbf{f}_{k+1}$ have corresponding arc lengths of $\lambda_f^{(j)}(k)$ and $\lambda_f^{(j)}(k+1)$ respectively.



p is the point on the line segment where **x** projects. Since the projection is orthogonal:

$$(\mathbf{x} - \mathbf{p}).(\mathbf{f}_{k+1} - \mathbf{f}_k) = 0 \tag{A1}$$

$$\text{Let } \mathbf{f}_{k+1} - \mathbf{f}_k = \boldsymbol{\Delta}_k \tag{A2}$$

$$\mathbf{p} = \mathbf{f}_k + \alpha_k(\mathbf{f}_{k+1} - \mathbf{f}_k) = \mathbf{f}_k + \alpha_k\boldsymbol{\Delta}_k \tag{A3}$$

Replacing equation A3 into equation A1,

$$(\mathbf{x} - \mathbf{f}_k - \alpha_k\boldsymbol{\Delta}_k).\boldsymbol{\Delta}_k = \mathbf{0} \Rightarrow \alpha_k = \frac{(\mathbf{x} - \mathbf{f}_k).\boldsymbol{\Delta}_k}{\|\boldsymbol{\Delta}_k\|^2} \tag{A4}$$

That leads to

$$\mathbf{p} = \mathbf{f}_k + \alpha_k\boldsymbol{\Delta}_k = \mathbf{f}_k + \left(\frac{(\mathbf{x} - \mathbf{f}_k).\boldsymbol{\Delta}_k}{\|\boldsymbol{\Delta}_k\|^2}\right)\boldsymbol{\Delta}_k \tag{A5}$$

and the resulting residual

$$\mathbf{r}(\mathbf{x},\mathbf{s_k}) = (\mathbf{x} - \mathbf{p}) = \left[\mathbf{x} - \mathbf{f_k} - \left(\frac{(\mathbf{x} - \mathbf{f_k}).\boldsymbol{\Delta}_\mathbf{k}}{||\ \boldsymbol{\Delta}_\mathbf{k}\ ||^2}\right)\boldsymbol{\Delta}_\mathbf{k}\right] \tag{A6}$$

The orthogonal distance from the line segment is then

$$d(\mathbf{x},\mathbf{s_k})=||\mathbf{r}(\mathbf{x},\mathbf{s_k})|| \tag{A7}$$

If the c^{th} line segment is the closest to the point **x**,

$$d(\mathbf{x},\mathbf{s_c}) = \min_k d(\mathbf{x},\mathbf{s_k}) \tag{A8}$$

The arc length corresponding to **x** is then

$$\lambda^{(\mathrm{j})}(\mathbf{x})=\lambda_{\mathrm{f}}^{(\mathrm{j})}(\mathrm{c})+\alpha_{\mathrm{c}}(\lambda_{\mathrm{f}}^{(\mathrm{j})}(\mathrm{c}+1)-\lambda_{\mathrm{f}}^{(\mathrm{j})}(\mathrm{c})) \tag{A9}$$

And the orthogonal distance from the principal curve is

$$d_{\mathrm{f}}^{(\mathrm{j})}(\mathbf{x})=d(\mathbf{x},\mathbf{s_c}) \tag{A10}$$

This derivation assumes that $0 \leq \alpha_c \leq 1$. If that condition is not met, we choose the line segment k that has the minimum orthogonal distance from **x** such that the corresponding α_{k} is from 0 to 1. At the extremities, there may be some points that will be orthogonal only to the extrapolated line segments. Our algorithm assigns those points to the line segment containing either the first or the last cluster center. It is possible (cf. figure 4.2) to have a point that is not orthogonal to any line segment. In that case, some error is tolerated by projecting it on the extrapolated line.

B. Interpolating Splines

Given n ordered pairs of points $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$, an interpolating spline is a collection of polynomials of the form

$$\begin{gathered} S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \\ for\ x_i < x \leq x_{i+1},\ i = 1,...,n - 1 \end{gathered} \tag{B1}$$

where the superscript i refers to the i^{th} polynomial in the spline.

The number of unknowns thus is 4(*n-1*) –or four parameters for each of the *n-1* segments. Clearly:

$$S_i(x_i) = y_i \text{ for } i = 1,...,n \tag{B2}$$

The smoothness requirements translate to the following conditions

$$[S_i(x_{i+1})]=[S_{i+1}(x_{i+1})] \text{ for } i = 1,...,n - 2 \tag{B3}$$

$$[S_i(x_{i+1})]'=[S_{i+1}(x_{i+1})]' \text{ for } i = 1,...,n - 2 \tag{B4}$$

$$[S_i(x_{i+1})]''=[S_{i+1}(x_{i+1})]'' \text{ for } i = 1,...,n - 2 \tag{B5}$$

Equations B2 through B5 provide *n*, (*n-2*), (*n-2*) and (*n-2*) degrees of freedom respectively. In order to find the parameters, 4(*n-1*)-(*n-1*)-3(*n-2*)=*2* more equations are required.

There are several choices for obtaining the two degrees of freedom. The second derivative for the first and *(n-1)*th polynomials can be set to zero. This produces a spline called the natural spline. The not-a-knot spline makes the third derivatives equal at the first and *(n-2)*th node. The not-a-knot spline is used in this dissertation, since there is no reason to believe that the spline should be linear at its endpoints.

Equations B3 through B5, in turn, give rise to the following algebraic equations:

$$a_i(x_{i+1} - x_i)^3 + b_i(x_{i+1} - x_i)^2 + c_i(x_{i+1} - x_i) + d_i = d_{i+1} \qquad \text{(B6)}$$

$$3a_i(x_{i+1} - x_i)^2 + 2b_i(x_{i+1} - x_i) + c_i = c_{i+1} \qquad \text{(B7)}$$

$$6a_i(x_{i+1} - x_i) + 2b_i = b_{i+1} \qquad \text{(B8)}$$

Equations B2, B6, B7 and B8 can be solved together with equation B9 provided by the not-a-knot condition, to obtain the spline parameters.

$$a_1 = a_2 \text{ and } a_{n+1} = a_{n+2} \qquad \text{(B9)}$$

It is easy to implement the code, but for convenience MATLAB program `spline.m` was used to find the parameters of the interpolating splines.

Vita

Sandeep Rajput was born on June 29th, 1975 in Kanpur, India. Notwithstanding frequent relocations and school changes, he managed to actively participate in sports and quizzes, and finished his school studies with exceptional honors. He joined the Indian Institute of Technology, Kanpur in July 1992 and graduated with a major in Chemical Engineering and a minor in Environmental Science and Engineering in May 1996. From 1996 to 1998 he worked with Reliance Industries Limited, Bombay as Assistant Manager (Projects) and provided technical services to mainly the Polyester plants in addition to his participation in a revamp project. He joined The University of Tennessee, Knoxville in July 1998. His graduate assistantship was provided by MCEC CANDIES Project. The project entailed nonlinear time series analysis of industrial data through MCEC member companies, and continued previous work to develop a GUI-based Nonlinear Time Series Analysis Software (NTSAS) in MATLAB®. During his five years in UT Knoxville, he enjoyed learning three foreign languages, traveling thrice to Europe, and avid reading in addition to writing and presenting his research, with the latter requiring considerable travel which was, of course, welcome.