DETECTING CHANGE IN NONLINEAR DYNAMIC PROCESS SYSTEMS

- Leon Christo Bezuidenhout -

Thesis submitted in partial fulfilment of the requirements for the degree Master of Science in Engineering (Chemical Engineering) in the Department of Process Engineering at the University of Stellenbosch

Study Leader: Prof Chris Aldrich

-March 2004-

DECLARATION

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: _____

Date: _____

SUMMARY

As result of the increasingly competitive performance in today's industrial environment, it has become necessary for production facilities to increase their efficiency. An essential step towards increasing the efficiency of these production facilities is through tighter processes control. Process control is a monitoring and modelling problem, and improvements in these areas will also lead to better process control.

Given the difficulties of obtaining theoretical process models, it has become important to identify models from process data. The irregular behaviour of many chemical processes, which do not seem to be inherently stochastic, can be explained by analysing time series data from these systems in terms of their nonlinear dynamics. Since the discovery of time delay embedding for state space analysis of time series, a lot of time has been devoted to the development of techniques to extract information through analysis of the geometrical structure of the attractor underlying the time series. Nearly all of these techniques *assume* that the dynamical process under question is stationary, i.e. the dynamics of the process did not change during the observation period. The ability to detect dynamic changes in processes, from process data, is crucial to the reliability of these state space techniques.

Detecting dynamic changes in processes is also important when using advanced control systems. Process characteristics are always changing, so that model parameters have to be recalibrated, models have to be updated and control settings have to be maintained. More reliable detection of changes in processes will improve the performance and adaptability of process models used in these control systems. This will lead to better automation and enormous cost savings.

This work investigates and assesses techniques for detecting dynamical changes in processes, from process data. These measures include the use of multilayer perceptron (MLP) neural networks, nonlinear cross predictions and the correlation dimension statistic.

The change detection techniques are evaluated by applying them to three case studies that exhibit (possible) nonstationary behaviour.

From the research, it is evident that the performance of process models suffers when there are nonstationarities in the data. This can serve as an indication of changes in the process parameters. The nonlinear cross prediction algorithm gives a better indication of possible nonstationarities in the process data; except for instances where the data series is very short. Exploiting the correlation dimension statistic proved to be the most accurate method of detecting dynamic changes. Apart from positively identifying nonstationary in each of the case studies, it was also able to detect the parameter changes sooner than any other method tested. The way in which this technique is applied, also makes it ideal for online detection of dynamic changes in chemical processes.

OPSOMMING

Dit is belangrik om produksie aanlegte so effektief moontlik te bedryf. Indien nie, staar hulle die moontlikheid van finansiële ondergang in die gesig – veral as gevolg van toenemende mededinging die industrie. Die effektiwiteit van produksie aanlegte kan verhoog word deur verbeterde prosesbeheer. Prosesbeheer is ‘n moniterings en modellerings probleem, en vooruitgang in hierdie areas sal noodwendig ook lei tot beter prosesbeheer.

Omdat dit moeilik is om teoretiese proses modelle af te lei, word dit al hoe belangriker om modelle vanuit proses data te identifiseer. Die ongewone optrede van baie chemiese prosesse, wat nie inherent stogasties blyk te wees nie, kan meestal verklaar word deur tydreeks data vanaf hierdie prosesse te analiseer in terme van hul nie-liniêre dinamika. Sedert die ontdekking van tydreeksontvouing vir toestandveranderlike stelsels, is baie tyd daaraan spandeer om tegnieke te ontwikkel wat inligting uit tydreekse kan onttrek deur die onderliggende geometriese struktuur van die attraktor te bestudeer. Byna al hierdie tegnieke aanvaar dat die dinamiese proses stationêr is, m.a.w dat die dinamika van die proses nie verander het tydens die observasie periode nie. Die vermoë om hierdie dinamiese proses veranderinge te kan identifiseer, is daarom baie belangrik.

Ook in gevorderde beheerstelsels is vroegtydige identifisering van dinamiese veranderinge in prosesse belangrik. Proses karakteristieke is altyd besig om te verander, sodat model parameters herkalibreer moet word, modelle opgedateer moet word en beheer setpunte onderhou moet word. Meer betroubare tegnieke om veranderinge in prosesse te identifiseer sal die aanpasbaarheid van proses modelle in hierdie beheerstelsels verbeter. Dit sal lei tot beter outomatisering en sodoende lei tot enorme kostebesparings.

Hierdie werk ondersoek tegnieke om dinamiese veranderinge in prosesse te identifiseer, deur die analise van proses data. Die tegnieke wat gebruik word sluit die volgende in:

multilaag-perseptron neurale netwerke, nie-liniêre kruisvoorspelling statistieke en die korrelasie dimensie statistiek. Die tegnieke is op drie gevallestudies toegepas om te sien of hulle die dinamiese veranderinge in die data kan identifiseer.

Vanuit die navorsing is dit duidelik dat proses modelle nadelig beinvloed word deur nie-stationêre data. Dit kan dien as ‘n indikasie van veranderinge in die proses parameters. Die nie-liniêre kruisvoorspellings algoritme gee ‘n beter indikasie van dinamiese veranderinge in die proses data, behalwe waar die tydreeks baie kort is. Toepassings van die korrelasie dimensie statistiek gee die beste resultate. Hierdie tegniek kon dinamiese veranderinge vinniger as enige ander tegniek identifiseer, en die manier waarop dit gebruik word maak dit ideaal vir die identifisering van dinamiese veranderinge in chemiese prosesse.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

1) My study leader, Prof Chris Aldrich.

2) Juliana Steyl, the secretary of Prof Chris Aldrich, for all the administrative work.

3) My fellow post graduate students, especially Gorden, for helping me to understand some of the theory behind nonlinear time series analysis.

4) JP Barnard and Prof C. Aldrich for using the Quickident Toolbox (Barnard & Aldrich, 2000) to do most of the data classification and state space reconstruction calculations.

5) Kevin Judd for using his code to do the correlation dimension calculations.

6) Rainer Hegger, Holger Kantz and Thomas Schreiber for using the *nonlinear cross prediction error* algorithm from the TISEAN package.

7) Macia, for always being supportive and understanding.

1 INTRODUCTION

There is no doubt that quality has become a major feature in the survival plan of companies. With diminishing markets resulting from the improved competitive performance in today's industrial environment, it is clear that unless there is a definite commitment towards increasing the efficiency of production facilities, they will loose their competitive edge. This will ultimately lead to elimination and the resultant harsh realities that come with it for the employees.

Improving the efficiency of production facilities can have a positive impact on chemical processes in several ways:

- Improvement in the quality of the final product
- Increase in production
- Decrease in the generation of hazardous wastes.

The first two issues are probably most important for a company and could mean the difference between success and failure. In previous times, decreasing hazardous waste was usually not considered a main objective. Recently, however, production facilities have come under pressure to comply with increasingly stringent environmental regulations, which makes this an import consideration. For the production facility, these pressures translate into a continuous effort to reduce process variability and maintain stability. This is usually accomplished through tighter process control, and is the reason why so much time and effort goes into developing advanced control systems.

One of the major problems with the use of advanced control systems is that process characteristics are always changing, so that model parameters have to be recalibrated, models have to be updated and control settings have to be maintained. More reliable detection of changes in processes will not only improve the performance and adaptability of process models used in control systems, but will also lead to faster

detection of dynamic changes and/or process upsets by monitoring these processes. This will lead to better automation and will result enormous cost savings.

The overall goal of this work is to investigate and develop suitable measures for detecting dynamic changes in processes from process data, and evaluate their performance.

1.1 Motivation

The degree of automation in chemical process control plants is still low. Many of the processes involve human operators. These operators are able to cope with the complexities of systems by applying their know-how skills acquired over years of hands-on operation of the plant. Given the complexity of the system and vagueness of situations, human operators are not always able to cope with the problem and provide good control performance. Control of manufacturing processes that are based on chemical reactions is often difficult in practice. Among the reasons are the nonlinear behaviour of such systems, large dead-times, and sometimes there are many conflicting goals within the system.

The control of chemical processes is a process monitoring and modelling problem. Improving monitoring and modelling techniques, will necessarily also lead to improved process control.

Process monitoring:

Through process monitoring it is possible to detect failing sensors and process upsets, which are important for reasons of safety and process efficiency. Process control based on faulty sensors is inefficient and can lead to unsafe operating conditions. Process upsets or disturbances can also lead to operating inefficiencies. An integral aspect of the overall process control problem is therefore timely identification of failed sensors and upsets. Apart from detecting failing sensors and process upsets, one also want to detect less trivial changes in the process dynamics. Doing so will further improve process monitoring, as it allows the identification of different dynamic operating regions in the process which can each be exploited separately.

Recent advances in process instrumentation and data collection techniques resulted in an increase in the amount of data recorded from chemical processes. Processes typically get more heavily instrumented, and measurements are done more extensively and more frequently. The result is a huge amount of process data. An increase in process data alone do not guarantee a better understanding of the process, and can be rather overwhelming and confusing. Much of the data may be redundant due to the high correlation of the measured variables. The difficulty lies in extracting only the most *significant* information from the plethora of the data. Unless this is done, monitoring and modelling the process can become problematic as result of the magnitude of data. This is also known as *information overload*. Ironically, most of the information is usually about less important process variables, and for the vital process variables (the ones that govern the process) there may be a lack of data.

Part of the research involves the use of advanced data analysis techniques to overcome these common problems.

Assessment of the validity of process models:

Modelling has become an increasingly important aspect of chemical process control. This is evident by the progression of controller design methods. The modern era of process control started by applying PID (*Proportional* + *Integral* + *Derivative*) controllers, also known as conventional controllers. PID controllers account for more than 80% of installed automatic feedback control devices in the process industries (Willis & Tham, 1994). There are several drawbacks using PID controllers, as they tend to operate within a limited operating range and are only optimal for a second order linear process without time delays. In practise, process characteristics are usually nonlinear and can change with time. PID controllers therefore do not cope well with nonlinearity in processes or changes in process parameters.

Recent advances in process control algorithms, particularly the model based controller design methods, have further increased the reliance on process models. Controller design advanced to using process models in the actual setting of the PID tuning parameters by implementing the controller within an adaptive framework (Willis & Tham, 1994). In this setup (***Figure 1.1***), the parameters of a model are updated regularly to reflect the current process characteristics. The controller settings can be updated continuously according to changes in the process characteristics. To ensure

optimal process control, rapid and reliable detection of changes in the process characteristics is vital. This is one area where improved change detection techniques will be particularly valuable.

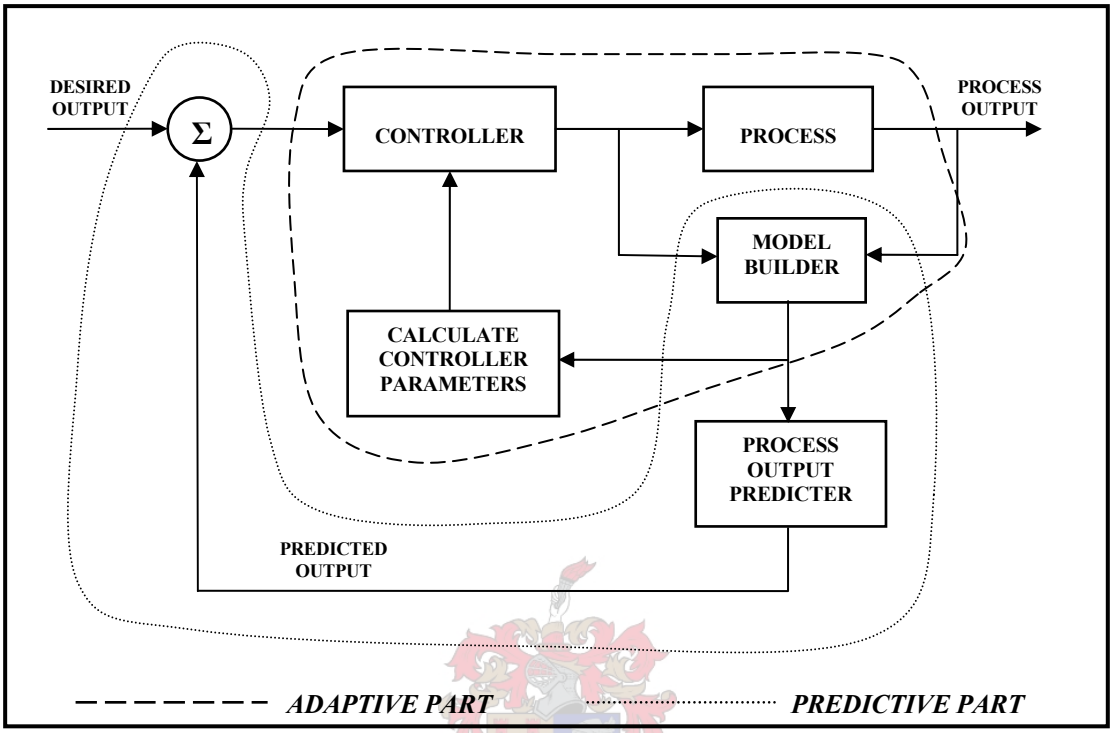


Figure 1.1: *Simplified Schematic of the Structure of Adaptive Predictive Controllers*

The model-based control strategy that has been most widely applied in the process industries is model predictive control (MPC) (Perry & Green, 1997). One major advantage of MPC is that it can accommodate difficult or unusual dynamic behaviour such as large time delays, nonlinearities and inverse responses. It is also well-suited for difficult multi-input, multi-output control problems where there are significant interactions between the manipulated inputs and the controlled outputs. A key feature of MPC is that future process behaviour is predicted using a dynamic model and available measurements from the process. The controller outputs are calculated to minimise the difference between the predicted process response and the desired response (***Figure 1.1***). At each sampling instance, the control calculations are repeated and the predictions updated based on current measurements from the process. MPC has the potential to provide "perfect" automatic control – that is ***if*** an ideal model of the process exists (Willis & Tham, 1994). This is why the critical factor in the successful

application of MPC (or any model-based technique for that matter) is the availability of a suitable dynamic model.

A model is nothing more than a mathematical abstraction of a real process (Seborg et al., 1989). The equations that comprise the model are at best an approximation to the true process. The model, therefore, cannot incorporate all of the features, both macroscopic and microscopic, of the real process. Depending on how they are derived, there are three different model classifications:

- Theoretical models developed using the principles of chemistry and physics.
- Empirical models obtained from a statistical analysis of process data
- Semi-empirical models that are a compromise between theoretical and empirical models

Theoretical models have several advantages over statistical/empirical models. They can often be extrapolated over a wider range of operating conditions than purely statistical/empirical models, which are only accurate over a limited range. Theoretical models also provide the capability to infer how unmeasured or unmeasurable process variables vary as the process operating conditions change, which gives a deeper understanding of the process.

The engineer normally has to seek a compromise involving the cost of obtaining the model, that is, the time and effort required to obtain and verify it. To establish a useful theoretical model, specialized knowledge about the system under study is needed. Theoretical models have parameters and fundamental relations that need to be evaluated from physical experiments. This can be time consuming and expensive. Furthermore, theoretical models are often compromised because they require so many simplifying assumptions in order to be tractable that they are often biased.

Given all the difficulties of obtaining theoretical models, it has become increasingly important to identify models from process data. Such models, which simply describe the functional relationships between the system inputs (*input space*) and system outputs (*output space*), are referred to as *black box models* (Juditsky et al., 1995). Although the parameters of these models do not have any physical significance in terms of equivalence to process parameters, such as heat or mass transfer coefficients and

reaction kinetics, the aim is merely to represent trends in the process behaviour faithfully. A problem frequently encountered when attempting to build *black box models*, is the availability of process data for all the variables in the input space. All the relevant input variables necessary to build the model are either just not recorded (probably as result of the costs involved), or are impossible to record. Most of the time, the engineer's only source of process data is that *single* output variable that needs to be controlled. Although such a modelling exercise might seem fundamentally flawed, a technique, called *state space reconstruction*[1], does exist to reconstruct the input space from this one-dimensional output. In this approach, the process data are viewed as a time series. Unlike the analyses of random samples of observations that are discussed in the context of most other statistics, the analysis of time series data is based on the assumption that successive values in the process data represent consecutive measurements taken at equally spaced time intervals.

The main assumption underlying time series analysis is that the properties or parameters describing the data, i.e. the dynamics governing the process, are constant (Schreiber, 1997). In other words, the dataset must be *stationary*. In practice, this is rarely true. Process parameters are often subject to changes at unknown time instants. Process parameters can also slowly drift with time, which is a signature of the existence of nonstationarity. Nonstationarities in the data will therefore complicate time series analysis and can result in a process model that is inferior or unreliable. The overall process model can be improved by detecting dynamic changes in the process data and dividing it into stationary segments that can each be modelled separately. These *sub-models* can be incorporated into a control strategy where the process is continuously being monitored for dynamic changes and where, depending on the process parameters, a different model is used for control. Once again, reliable methods that are sensitive to dynamic changes are vital for such an implementation.

[1] The theory behind state space reconstruction is thoroughly explained in ***Chapter 2***.

1.2 Goals, Scope and Approach

The problem of detecting changes in properties of signals and systems has received increasing attention in the last twenty or so years. Most of the work, however, has been done on linear stochastic systems (Basseville & Nikiforov, 1993). Change detection for nonlinear dynamical systems has not been investigated in any depth. This was largely due to a historically poor understanding of the subject. In the last few years, with computers being more accessible and processing power becoming increasingly cheaper, methods for analyzing nonlinear time series have seen dramatic progress. These advances now make it possible to investigate more complex areas of interest, such as detecting dynamic changes in nonlinear systems, in much more detail.

The main objective of this work is to detect changes (or nonstationarities) in dynamic process systems from process data. It includes changes that can occur both relatively fast or relatively slow with respect to the sampling period of the measurements. These changes by no means imply changes which are large in magnitude. The key difficulty is to detect intrinsic changes that are not necessarily directly observed and that are measured together with other types of perturbations. A key part of the research is to identify suitable methods for detecting changes in process systems, and evaluate their ability to detect changes quickly and reliably.

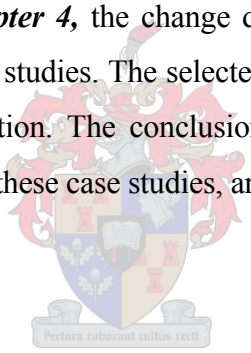
The research is focused on nonlinear dynamic systems that exhibit *deterministic* behaviour, or at least mixed systems that have a *dominant deterministic* part. The theory behind nonlinear time series analysis forms an integral part of the research, and most of the work is done within this context.

The change detection problem is approached by first classifying the data. It is important to determine whether the time series data are linear or nonlinear, and whether it exhibits stochastic or deterministic behaviour. This classification is a complicated part of time series analysis. Usually the data do not belong purely to a specific class and are therefore classified according to the degree of its behaviour. *Surrogate data analysis* techniques will be used to classify the system. The outcome of this classification will indicate whether dynamic change detection techniques, as discussed within the context of this work, need to be applied, or whether traditional linear statistics would be sufficient to detect changes in the process. The next step will be to apply the identified

change detection techniques to various case studies consisting of simulated, as well as actual data from chemical processes. The techniques will then be evaluated on their ability to detect different kinds of dynamic change, as well as the time in which they could detect the dynamic change.

1.3 Thesis Layout

All the relevant issues concerning the state space analysis of time series are discussed in ***Chapter 2***. This includes the theory behind state space reconstruction, surrogate data analysis, as well as various other advanced analysis techniques and statistics that are needed to quantify dynamic behaviour in processes from time series data. This theory is fundamental to the research and receives a great deal of attention. ***Chapter 3*** focuses on the issue of dynamic changes in processes, and measures for detecting these changes are identified and discussed. In ***Chapter 4,*** the change detection methodology is put into practice and applied to three case studies. The selected change detection techniques are investigated in depth in this section. The conclusions and limitations of the change detection methodology, based on these case studies, are discussed in ***Chapter 5.***

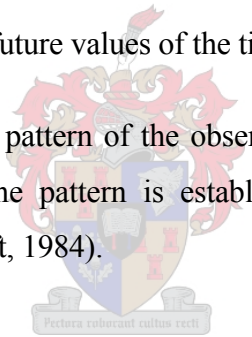


2 STATE SPACE ANALYSIS OF TIME SERIES

Time series data are sequences of measurements that follow non-random orders. Unlike the analyses of random samples of observations that are discussed in the context of most other statistics, the analysis of time series is based on the assumption that successive values in the data represent consecutive measurements, usually taken at equally spaced time intervals. There are two main goals for analyzing time series data:

- Identifying the nature of the phenomenon represented by the sequence of observations
- Forecasting – predicting future values of the time series variables.

Both these goals require that the pattern of the observed time series data is identified and formally described. Once the pattern is established, it can be interpreted and integrated with other data (Statsoft, 1984).



2.1 Dynamical Systems

Time series analysis, and especially nonlinear time series analysis, are motivated and based on the theory of ***dynamical systems***; that is, the time evolution is defined in some phase space (Kantz & Schreiber, 1997). Since these systems can exhibit deterministic chaos[1], this is a natural starting point when irregularity is present in a signal.

A purely deterministic system is defined as a system where, once its present state is fixed, the states at all future times are determined as well. It is therefore essential to establish a vector space for the system. Such a ***state space*** or ***phase space*** specifies the

[1] *Chaos – Irregular but deterministic motion which is characterized by a continuous, broadband Fourier spectrum. Possible only in a three-or-more dimensional nonlinear system of differential equations or a two-or-more dimensional nonlinear discrete time map. This nonlinear motion is slightly predictable, non-periodic and sensitive to changes in initial conditions. (Abarbanel, 1998)*

state of the system by specifying a point in the system. The construction of a state space makes it possible to study the dynamics of the system by studying the dynamics of the corresponding state space points. In theory, dynamical systems are generally defined by a set of first order ordinary differential equations acting on a state space. If certain conditions are met, the mathematical theory of ordinary differential equations ensures the existence and uniqueness of the trajectories[2] in the state space.

In a deterministic dynamical system, the state space is assumed to be a finite-dimensional vector space, $\Re^d$, where a state is specified by a vector $\boldsymbol{x} \in \Re^d$. The general form for an autonomous discrete-time dynamical system is the map

$$x_{n+1} = F(x_n), \qquad n \in \mathbb{Z} \tag{2.1}$$

where $\boldsymbol{x}_n, \boldsymbol{x}_{n-1} \in \Re^d$ and $\boldsymbol{F} : \Re^d \to \Re^d$ is a diffeomorphism[3]. In the second case, time is a continuous variable:

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \qquad t \in \mathbb{R} \tag{2.2}$$

This form is normally referred to as flow. The flow is called autonomous if f does not explicitly depend on time (t).

A sequence of points ($\boldsymbol{x}_n$ or $\boldsymbol{x}(t)$) solving the above equations is called a trajectory of the dynamical system, where the initial condition is $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Depending on the initial condition and the form of F (or f), the trajectory will either run away to infinity as time proceeds or stay in a bounded area forever. In a dissipative[4] system, the points visited by the system after transient behaviour has died out will be concentrated on a subset of state space. The geometric object to which the trajectory orbits go in time is called the system attractor[5]. ***Figure 2.1*** gives an illustration of a typical attractor. In this particular case, the attractor is three dimensional and each axis represents one of the variables by which the system is described.

---

[2] *Trajectory – The path that a signal follows through state space*

[3] *Diffeomorphism – A smooth mapping with a smooth inverse.*

[4] *Dissipative system – A system with sources and sinks of energy*

[5] *Attractor – The set of points in state space visited by a signal trajectory after the transients are gone*

It is important to note that even for non-deterministic systems the concept of the *state of a system* is still very powerful.

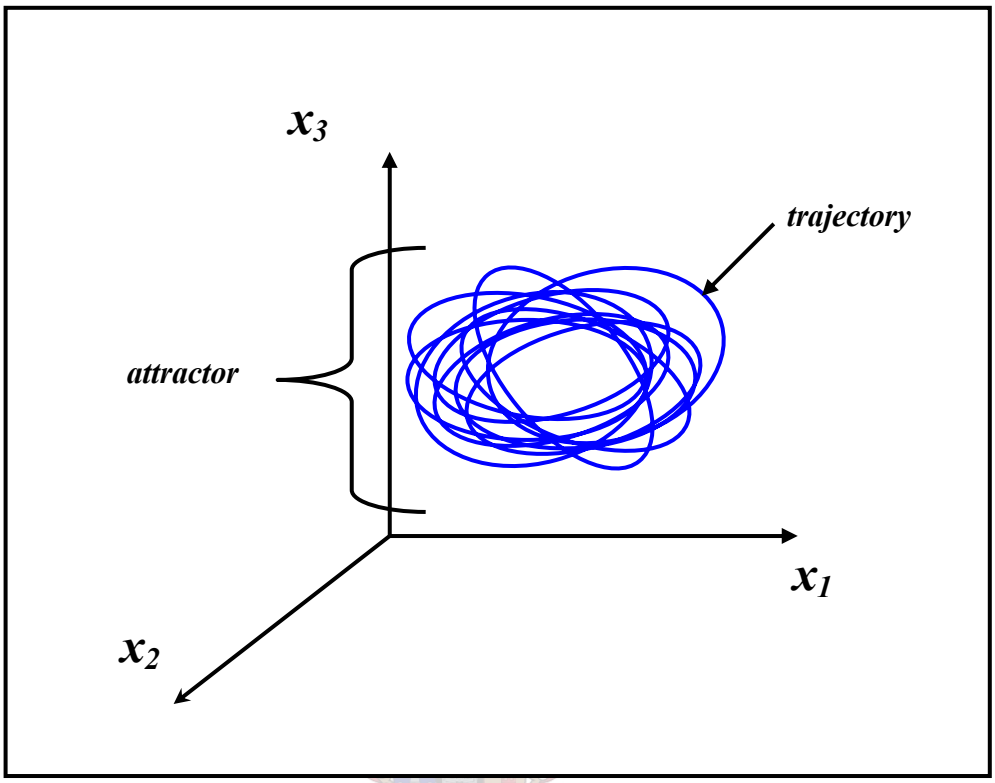


Figure 2.1: *Illustration of a three dimensional attractor, where* $\boldsymbol{x_1, x_2, x_3}$ *are the three variables governing the system.*

2.2 Linear Systems

Consider the case where the flow in ***Equation (2.2)*** is linear. An autonomous linear system of f degrees of freedom, $\boldsymbol{x}(t) = [x_1(t), x_2(t), ..., x_f(t)]$, would yield the following equation (Abarbanel, 1993):

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{A}.\boldsymbol{x}(t) \tag{2.3}$$

Where $\boldsymbol{A}$ is a constant $f \times f$ matrix. The possible courses of evolution of ***Equation (2.3)*** are characterized by the eigenvalues of the matrix $\boldsymbol{A}$. The trajectories of the solution behave in one of the following ways:

1) Directions in *f*-space along which the orbits shrink to zero – namely, directions along which the real part of the eigenvalues of $\boldsymbol{A}$ are negative, or

2) Directions along which the orbits unstably grow to infinity – namely, directions along which the real part of the eigenvalues are positive, or

3) Directions where the eigenvalues occur in complex-conjugate pairs along with zero or negative real part. When the eigenvalues have an absolute value of unity, the trajectories will follow closed circular or elliptical orbits (Honkela, 2001). However, in any situation where the eigenvalues has a positive real part, it is an indication that the linear dynamics are incorrect, and one must go to the nonlinear equations that govern the process to make a better approximation to the dynamics (Abarbanel, 1993).

The linear case can be illustrated by means of a simple example, using the following two coupled linear differential equations:

$$\frac{dx}{dt} = -\omega y, \qquad \frac{dy}{dt} = \omega x \tag{2.4}$$

The equations have a periodic solution in the form (Kantz & Schreiber, 1997):

$$x(t) = a.\cos[\omega(t - t_0)], \qquad y(t) = a.\sin[\omega(t - t_0)] \tag{2.5}$$

When inspecting ***Equation (2.5),*** as well as the visual illustration thereof (***Figure 2.2*** and ***Figure 2.3*****),** it is clear that the solution will stay finite forever. This demonstrates the fact that autonomous linear dynamic systems are too straightforward to describe any interesting phenomena. In practice, stable linear systems either exponentially converge to a constant value, or exhibit periodic behaviour.

Linear time series analysis is therefore well defined (Statsoft, 1984), and established linear mathematical techniques sufficiently meet the requirements.

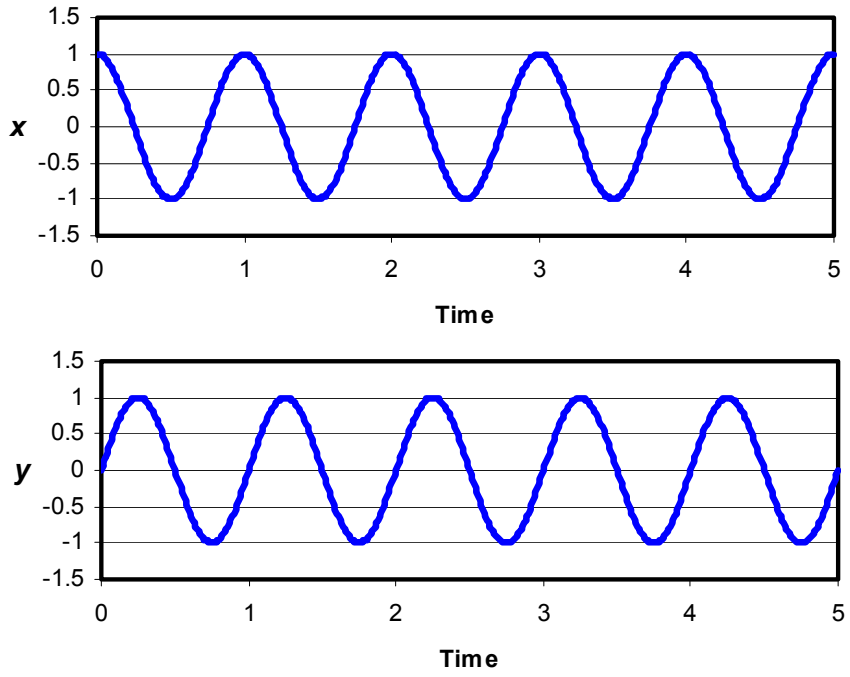


Figure 2.2: *Plot of* ***x*** *and* ***y*** *vs.* ***time*** *showing periodic nature of the solution*

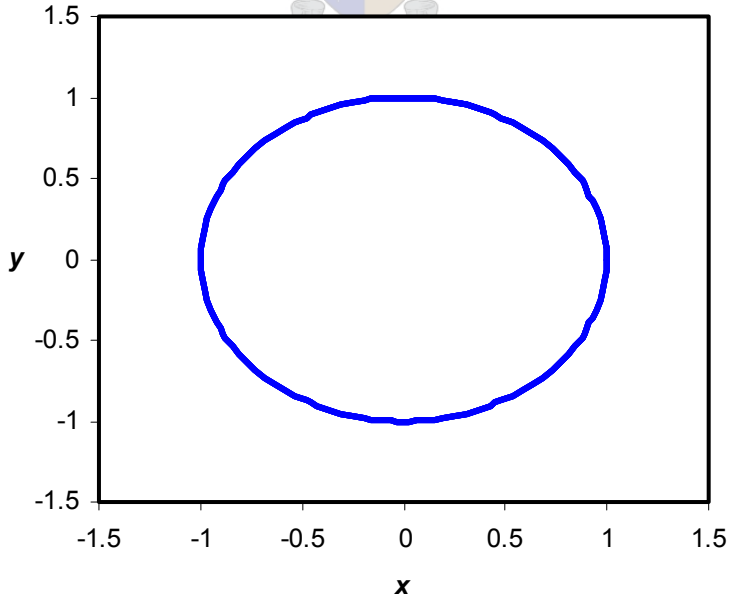


Figure 2.4: *Plot of* ***x*** *vs.* ***y*** *showing the closed circular orbit of the attractor.*

2.3 Nonlinear Systems and Chaos

While the possible dynamics of linear systems are rather restricted, even very simple nonlinear systems can have very complex dynamical behaviour. Previously people believed that only a stochastic[6] or noisy input to the system could create a stochastic output, and that only a deterministic input to a deterministic system created well-behaved deterministic outputs. In addition, it was believed that a small change in the initial conditions of the dynamic equations created only a small change at any future time. It has now become common knowledge that this is not true for nonlinear dynamical systems (see ***Figure 2.4***). Deterministic input to a deterministic dynamical system can create a stochastic or irregular noise-like chaotic output, and a small change in the initial conditions can lead to an entirely different output after some lapse of time. (Chang & Lee, 1996).

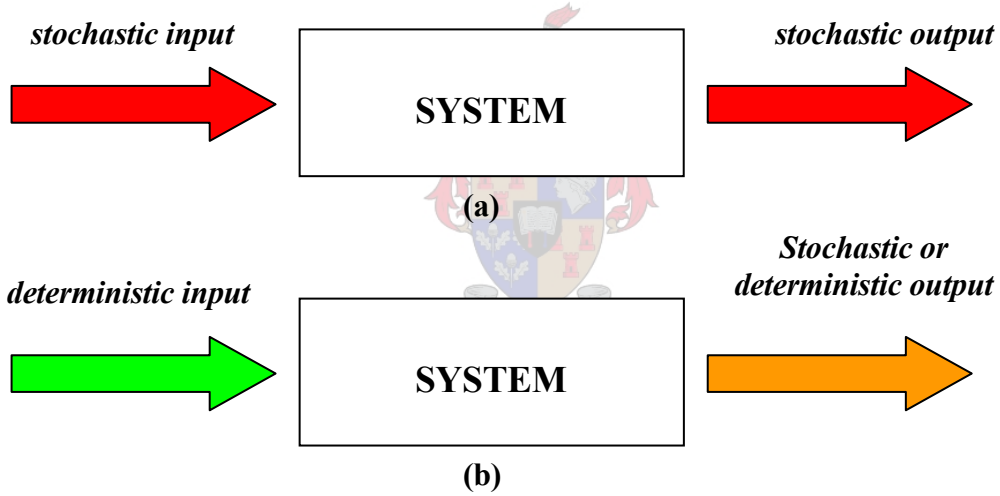


Figure 2.4: a) *Conventional concept of system behaviour;* ***b)*** *Actual system behaviour*

Not all irregular motions are chaos. Irregularity may be caused by some other underlying reasons. For a process to be chaotic it must satisfy a series of tests. The most important test is that the signal has a sensitive dependence on initial conditions. It

[6] *Stochastic - Stochastic is synonymous with "random". Opposite of "deterministic"*

should also be indecomposable or ergodic[7]; and have an element of regularity (Eckmann & Ruelle, 1985).

Chaos seems to be the rule, rather than the exception, in nature. It also frequently occurs in process systems engineering, where nonlinear complexity and a great number of describing equations govern the dynamics. This is especially true for chemical engineering, where chemical processes are inherently nonlinear and often have structurally unstable dynamics. Although chaotic behaviour often occurs in chemical engineering processes, it is usually attributed to process noise and treated as such. The statistical approach normally used in such situations is without doubt a powerful tool. However, in process systems engineering there has to be dealt with interactions of highly complex nonlinear phenomena with multiple units involving chemical reactions, heat and mass transfer, separations, fluid flow, etc. Since chaos can occur without any single noise input to any single unit, the complicated chaos resulting from all the individual units, integrated as a system, is just imaginable (Chang & Lee, 1996).

Chaotic behaviour can be illustrated visually using the Hênon-map as an example:

$$x_{n+1} = a - x_n^2 + by_n, \qquad y_{n+1} = x_n \tag{2.6}$$

The map yields irregular solutions for many choices of ***a*** and ***b***. Setting ***a* = *1.4*** and ***b* = *0.3***, for example, generates a typical sequence of x_n that will be ***chaotic*** (Kantz & Schreiber, 1997). The irregular signals are shown in ***Figure 2.5.*** Especially note the interesting shape of the chaotic attractor (***Figure 2.6***).

This illustration of the chaotic behaviour of the Hênon-map, verifies the earlier claim that *even simple nonlinear systems can have very complex dynamical behaviour.* It is therefore understandable that traditional linear analysis tools can usually not be applied to nonlinear, chaotic systems with great success. Doing so would be much harder and conclusion drawn from the results would be fundamentally limited in range.

[7] *Ergodic theory says that a time average equals a space average, where the weight with which the space average has to be taken is an invariant measure*

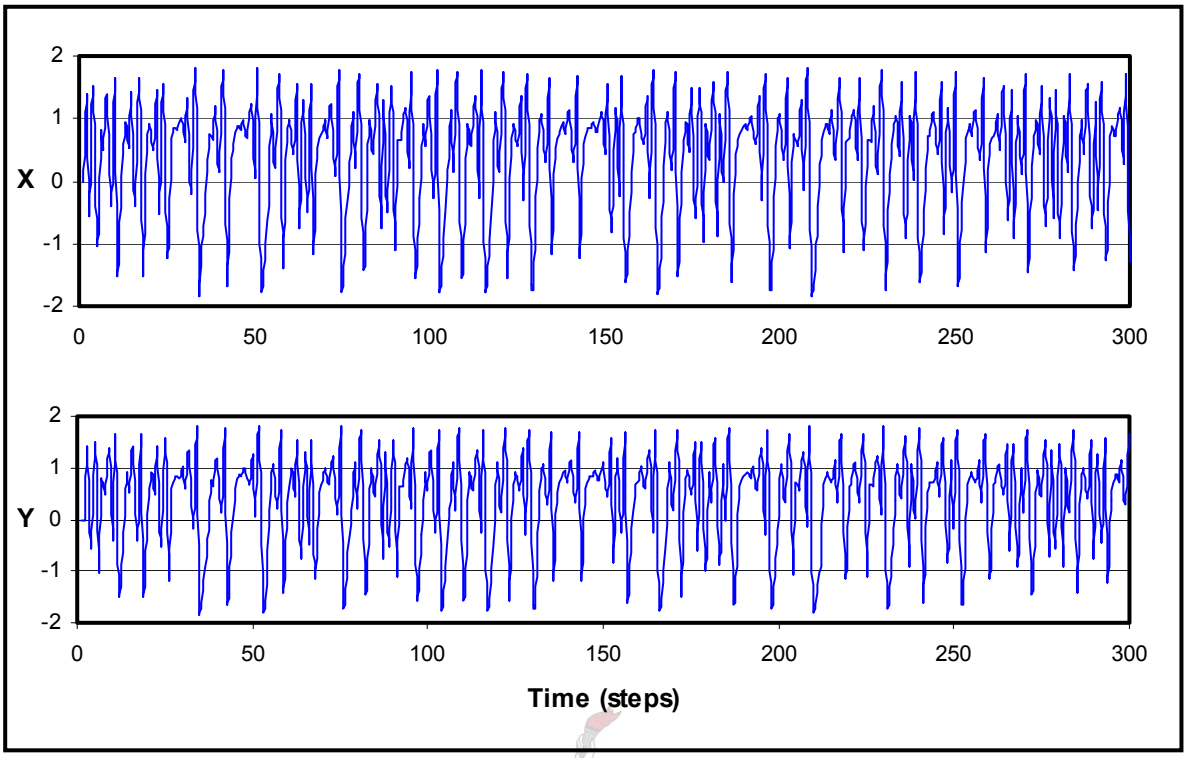


Figure 2.5: *Chaotic x,y signals of the Hênon-map.*

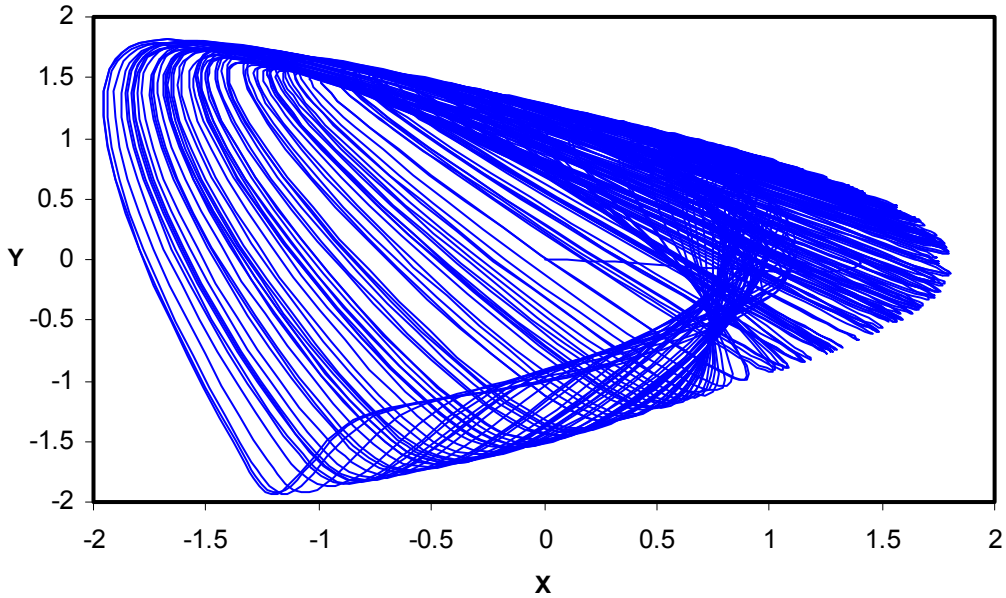


Figure 2.6: *Chaotic attractor of Hênon-map*

This predicament started the era of nonlinear time series analysis. Although nonlinear data analysis techniques are somewhat unconventional and often more difficult to use, it definitely is a big improvement on traditional methods. Even though long-term prediction of chaotic systems is impossible, it is often still possible to predict statistical features of its behaviour and find certain invariant[8] features that provide a qualitative description of the system.

2.4 State-Space Reconstruction

Apart from nonlinear and chaotic behaviour, intelligent data analysis problems are often complicated by the following, seemingly contradictorily, situation (Bradly, 1996): simultaneous overabundance and lack of data. Take for example the following chemical reactor setup:

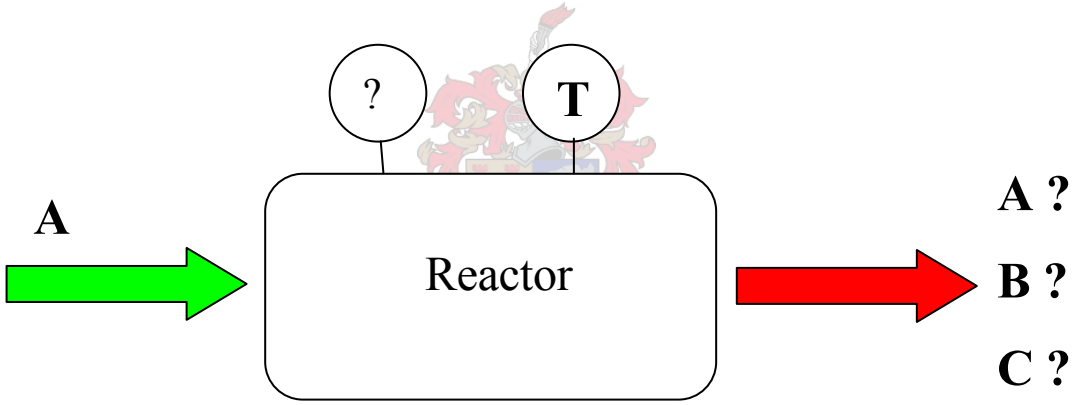


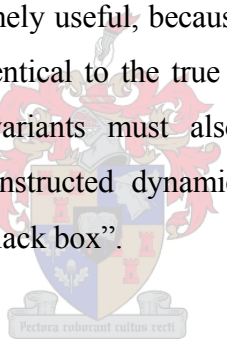
***Figure 2.7**: Example of a chemical reactor to explain the reasoning behind state space reconstruction.*

The dynamics of this system are governed by, amongst others, the temperature inside the reactor, the pressure inside the reactor and the concentration of the species A, B and C. If there were only a temperature sensor installed on the reactor, one would have loads of information available about the temperature inside the reactor, but no data about other important quantities such as species concentration or pressure. This is

[8] *Invariant - Unchanged by a specified transformation or operation*

usually the case when the other system properties are not sensor-accessible or hard to measure with available sensors. When doing intelligent data analysis, the analyst is therefore often required to extract meaningful conclusions about a complicated system using data from a single sensor. At first glance, the data analysis procedure would appear fundamentally limited.

Probably the most powerful, and widely used, way of getting around this problem is a technique called *time delay embedding*. Time delay embedding is a method of reconstructing the internal dynamics of a complicated (nonlinear) system from a single time series. That is, time delay embedding can often be used to infer important information about unmeasurable variables, such as the species concentration or reactor pressure in ***Figure 2.7***, from a single measured variable, e.g. the reactor temperature. The reconstruction produced by time delay embedding is not completely equivalent to the internal dynamics of the system in all situations. However, a properly done single sensor reconstruction can be extremely useful, because the results are guaranteed to be topologically (i.e. qualitatively) identical to the true internal dynamics of the system, and therefore the dynamical invariants must also be similar. This means that conclusions drawn from the reconstructed dynamics are also true of the internal unmeasured dynamics inside the "black box".



2.4.1 Embedding Theorems

The solution to the problem of how to go from scalar observations to multivariate state space is contained in the geometric theorem, called the *embedding theorem*, attributed to Takens (1981):

Let $\boldsymbol{M}$ be a smooth (C^2) m-dimensional manifold that constitutes the original state space of the dynamical system under investigation, and let $\phi^t : M \to M$ the corresponding flow. Suppose that it is possible to measure some scalar quantity $s(t) = h(\boldsymbol{x}(t))$ that is given by the measurement function $h : M \to \Re$ where $\boldsymbol{x}(t) = \phi^t(\boldsymbol{x}(0))$. It is then possible to construct a delay coordinate map that maps a state $\boldsymbol{x}$ from the original state space M to a *reconstructed state space* $\Re^{d_e}$:

$$F : M \to \Re^{d_e} \qquad (2.7)$$
$$\boldsymbol{x} \to \boldsymbol{y} = F(\boldsymbol{x}) = (s(t), s(t+t_l), s(t+2t_l), ..., s(t+(d_e-1)t_l)$$

Here d_e is the *embedding dimension* and t_l is the *lag* or *time delay* used.

Takens proved that for $d_e \geq 2m+1$ it is a generic property of F to be an embedding of M in $\Re^{d_e}$, that is $F : M \to F(M) \subset \Re^{d_e}$ is a C^2- diffeomorphism. Generic implies that the subset of pairs (h, t_l) is an open and dense subset in the set of all pairs (h, t_l). The theorem was later generalized by Sauer, York and Casdagli (1991) in two ways:

1) They replaced the condition $d_e \geq 2m+1$ by $d_e \geq 2d_0(A)$, where $d_0(A)$ is the box-counting dimension of the attractor $A \subset M$.

2) They replaced the term *generic* by the term *prevalent*, which means that *almost all* (h, t_l) will give an embedding.

The first improvement is great progress for experimental systems that have a low-dimensional attractor (e.g. $d_0(A) < 5$) in a very high dimensional (e.g. m=100) space. In this case Takens' theorem says that only for very large embedding dimensions (e.g. $d_e \geq 201$) diffeomorphic equivalence can be expected, whereas Sauer et al. (1991) says that a small ($d_e > 10$) embedding dimension will be adequate.

The second modification was necessary because examples of *open* and *dense* (i.e. generic) sets were found that were rather *thin* (Sauer et al., 1991). They also showed that for dimension estimation an embedding dimension $d_e > d_0(A)$ should be enough.

Assume that the scalar signal $s(t) = h(\boldsymbol{x}(t))$ is sampled with a sampling time, t_s. The resulting time series $\{s_n\}$ with $s_n = s(nt_s)$ is used to reconstruct the states

$$\boldsymbol{y_n} = (s_n, s_{n+l}, s_{n+2l}, ..., s_{n+(d_e-1)l}) \qquad (2.8)$$

for $n = 1,...,N$. The symbol l represents the delay time (*lag*) in units of the sampling time, $t_l = lt_s$ (Parlitz, 1995).

The $\boldsymbol{y_n}$ replace the scalar data measurements s_n with data vectors in a Euclidian d-dimensional space in which the invariant aspects of the sequence of points $\boldsymbol{x_n}$ are captured with no loss of information regarding to the properties of the original system. The newly reconstructed space is related to the original $\boldsymbol{x_n}$ - space by smooth, differential transformations. The smoothness is necessary in allowing the demonstration that all the invariants of the motion in the reconstructed space are the same as if they were evaluated in the original space. This suggests that just as much can be learned about a system studying the reconstructed state space, than studying the original (true) state space (Abarbanel, 1996). And herein lies the significance of state space reconstruction. A numerical example explaining the theory behind time delay embedding is given in ***Appendix A***.

2.4.2 Estimating Suitable Reconstruction Parameters

State space reconstruction depends on two parameters: the *lag* (l) and the embedding dimension (d_e). For the reconstruction to be useful (e.g. for dynamic modelling), it is important to choose suitable (optimal) values for these parameters. The following sections discuss methods to estimate the embedding parameters.

A. Choosing time delays

The embedding theorem states that any time lag is acceptable when reconstructing a state space. This, however, is not very useful when extracting dynamical information from the data. Different lags lead to reconstructions of the attractor that are diffeomorphically equivalent but geometrically different. If the lag (l) is too small, the coordinates $\boldsymbol{y_n} = s_{n+(i-1)l}$ will be so close to each other numerically, that they will be indistinguishable. This, from any practical point, will not provide two independent coordinates. On the other hand, when the lag is too large, the coordinates will be

statistically completely independent of each other and the projection of an orbit on the attractor is into two totally unrelated directions (Abarbanel, 1993).

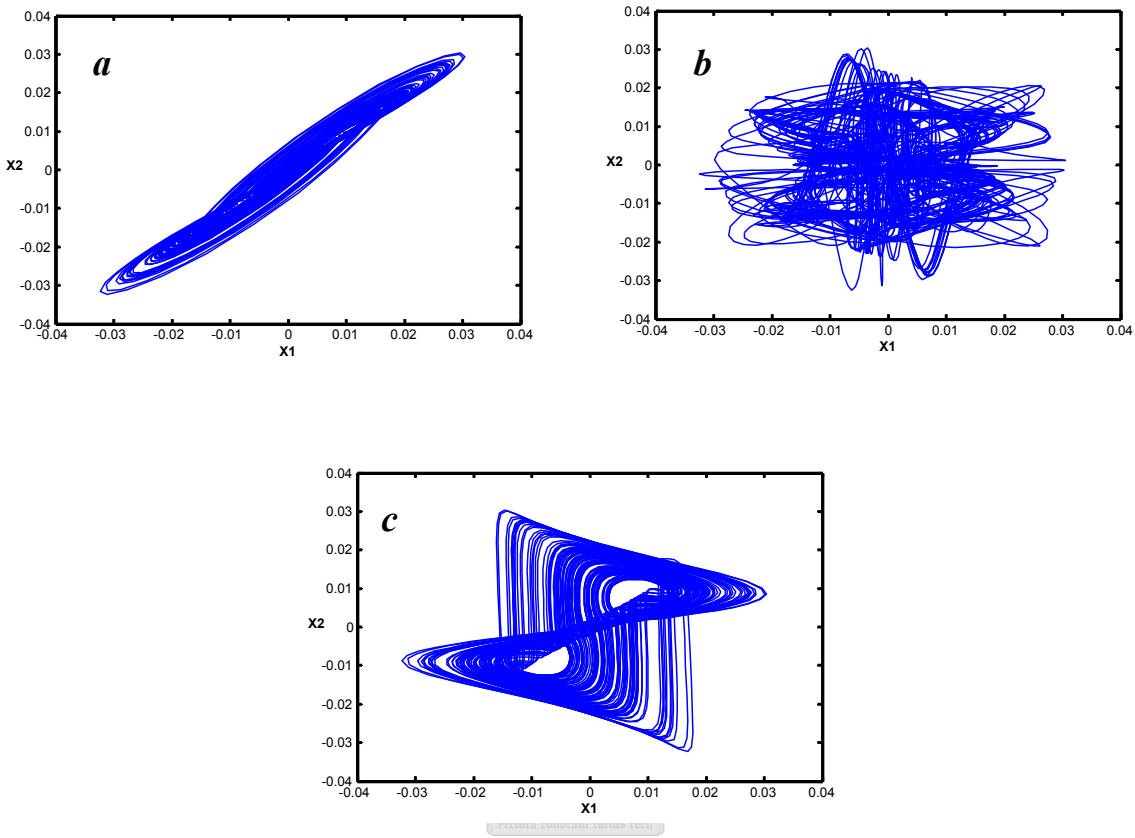


Figure 2.8: *Illustration of a) too small, b) too large and c) optimal time delay.*

The fundamental issue is that there must be a balance between values of l that are too small, where each component of the vector does not add significant new information about the system dynamics, and values of l that are too large (***Figure 2.8***) . Large values of l create uncorrelated elements in $\boldsymbol{y}_n$ because of the instabilities in the nonlinear system that becomes noticeable over time. This will result in the components of $\boldsymbol{y}_n$ becoming independent and convey no knowledge about the systems dynamics (Abarbanel, 1998).

It is therefore essential to determine an intermediate value for the lag that will give an optimal embedding. The *linear autocorrelation function* and the *average mutual information* (AMI) statistic are well-accepted methods for such calculations.

Linear Autocorrelation Function:

The *linear autocorrelation function* is defined as

$$C_L(l) = \frac{\frac{1}{N}\sum_{k=1}^{N}\left[s_{k+l} - \overline{s}\right]\left[s_k - \overline{s}\right]}{\frac{1}{N}\sum_{k=1}^{N}\left[s_k - \overline{s}\right]^2} \tag{2.9}$$

where

$$\overline{s} = \frac{1}{N}\sum_{k=1}^{N} s_k$$

The function gives information about the linear dependency of coordinates on each other. Higher values of C_L indicate a higher average linear correlation between coordinates at that specific lag (l).

Determining the time lag where $C_L(l)$ first passes through zero will give a good estimate of l. Choosing l to be the first zero of the function $C_L(l)$ would, on average over the observations, make the coordinates $[s_n, s_{n+l}, s_{n+2l}, \ldots, s_{n+(d_e-1)l}]$ linearly independent. It is important to note that this independency may have no relation to their nonlinear independence or their usefulness as coordinates of a nonlinear system. The approach is not a perfect solution to the problem, but it at least gives an indication of what lag (l) to choose.

Average Mutual Information:

While it is helpful to use linear dependence as criteria to determine optimal lag (l), it is preferable to use a measure that takes the aspect of chaotic behaviour into consideration.

Average mutual information (Frasier & Swinney, 1986) is such an approach and uses the concept of *information theory* to determine an optimal embedding lag.

Mutual information is a way of identifying how much information one can learn about a measurement at one time, from a measurement taken at another time. Say there exist two sets of measurements, set $A = \{a_i\}$ and set $B = \{b_j\}$, with a probability distribution associated with each system governing the possible outcomes of observations on them. The *mutual information* between measurement a_i drawn from set $A = \{a_i\}$ and measurement b_j drawn from set $B = \{b_j\}$, is the amount of information (in *bits*) about measurement a_i that is acquired by observing b_j. Mathematically this is represented by

$$I_{AB}(a_i, b_j) = \log_2 \left[\frac{P_{AB}(a_i, b_j)}{P_A(a_i) P_B(b_j)} \right] \tag{2.10}$$

where $P_{AB}(a,b)$ is the joint probability density for measurements from set A and B resulting in values a and b. $P_A(a)$ is the probability of observing a out of the set of all A and $P_B(b)$ the probability of finding b in a measurement from set B. The quantity $I_{AB}(a_i, b_j)$ is called the *mutual information* of the two measurements a_i and b_j, and is symmetric in how much is learned about b_j from measuring a_i. In a deterministic system these probabilities are evaluated by constructing a histogram of the variations of the a_i or b_j seen in their measurements.

If the measurement of a value from set A (resulting in a_i) is completely independent of the measurement of a value from set B (resulting in b_j), then $P_{AB}(a,b)$ factorizes to $P_A(a)P_B(b)$ and the amount of information between the measurements, $I_{AB}(a_i, b_j)$, is zero. The *average mutual information* between measurements of any value a_i from set A and b_j from B is the average over all possible measurements of $I_{AB}(a_i, b_j)$:

$$I_{AB} = \sum_{a_i, b_j} P_{AB}(a_i, b_j) I_{AB}(a_i, b_j) \tag{2.11}$$

This quantity is not related to the linear or nonlinear evolution rules of the quantities measured. It is strictly a set theoretical idea which connects two sets of measurements with each other and establishes a criterion for their mutual dependence based on the concept of information connection between them (Abarbanel, 1996).

To place this definition in the context of observations from a physical system, the set of measurements s_n is considered set *A*; and the set of measurements s_{n+l}, a time lag l later, as set *B*. The average amount of information about s_{n+l} that is acquired when making an observation of s_n, or in other words, the *average mutual information* between observations s_n and s_{n+l}, is then

$$I(l) = \sum_{n=1}^{N} P(s_n, s_{n+l}) \log_2 \left[\frac{P(s_n, s_{n+l})}{P(s_n) P(s_{n+l})} \right] \tag{2.12}$$

and $I(l) \geq 0$.

As is the case with the *linear autocorrelation function*, the *average mutual information* function is used to determine a time lag (l) when the values of s^n and s^{n+l} are independent enough of each other to be useful as coordinates in a time delay vector, but not so independent as to have no connection with each other at all. Fraser and Swinney (1986) suggest, as a prescription, to choose the lag where the first minimum of $I(l)$ occurs, as the lag (l) for the time delay reconstruction of the state space. This is a simple rule derived from their more detailed suggestion, but it serves quite well.

The average mutual information is in fact a kind of generalization from the correlation function in the linear world to the nonlinear world (Abarbanel, 1993 & 1996).

B. Choosing the embedding dimension

Choosing optimal values for the embedding dimensions is just as important as choosing optimum time lag (l) values. If the chosen embedding dimension is too small, the conditions given in the embedding theorems are not satisfied. On the other hand, if the

dimension is too large, practical problems occur due to the fixed amount of points that constitute thinner and thinner sets in $\Re^d$ as d is increased (Parlitz, 1995).

When considering possible values for embedding dimension, emphasis is placed on determining the integer global dimension where the number of coordinates chosen is just enough to unfold observed orbits from self-overlaps arising as result of projection of the attractor into a lower dimensional space. This means that if two points (or trajectory orbits) of a particular observation set lie close to each other in some dimension d they should do so because it is a property of the set of observations and not because of the small value of d in which the observations is viewed. The lowest dimension that unfolds the attractor so that no overlaps remain is called the embedding dimension (d_e). In practice, it is possible to guess a suitable value for d_e by successively embedding into higher dimensions and looking for consistency in the results. (Abarbanel, 1993)

From the previous, it should be clear that embedding data into a higher than necessary dimension does not create any ambiguity. The question now arises: why not always embed the observations into an arbitrary large dimension to be sure the attracter is totally unfolded? Two problems arise when working with dimensions that are larger than really required by the data and time delay embedding:

1) Computational time for extracting interesting properties from the embedded attractor increases exponentially with increasing d.

2) In the presence of noise or other high dimensional contamination of the data, the extra dimensions are not populated by dynamics already captured by a smaller dimension, but entirely by the contaminated signal. In an embedding space that is too large, unnecessary time is spend working around aspects of badly represented observations which are solely filled with noise.

This realization motivated the search for techniques that can identify an optimal embedding dimension from the data itself.

False Nearest Neighbours:

The *false nearest neighbours* technique developed by Kennel et al. (1992) is based on the following idea. For any point, $\boldsymbol{y}_n$, on an attractor one can ask whether its nearest neighbour, $\boldsymbol{y}_n^{NN}$, in a state space of dimension *d* is there for dynamical reasons, or if it is instead projected into the neighbourhood because the dimension is too low. That neighbour is then examined in dimension *d+1* by simply adding another coordinate to $\boldsymbol{y}_n$ using the time delay reconstruction. If the nearest neighbour under examination remains a neighbour in the larger space, it is a true neighbour which arrived there dynamically. If the neighbour moves away from the point $\boldsymbol{y}_n$ as dimensions are added, it was a false nearest neighbour and only a neighbour because the dimension *d* was too low. Once the number of false nearest neighbours becomes zero, he attractor has been ambiguously unfolded because all crossings of the orbits have been eliminated.

The closeness of two points in the state space is determined by the Euclidean distance between them. In a $\Re^d$ - embedding, each point

$$\boldsymbol{y}_n = (s_n, s_{n+l}, s_{n+2l}, \ldots, s_{n+(d-1)l}),$$

has a nearest neighbour

$$\boldsymbol{y}_n^{NN} = (s_n^{NN}, s_{n+l}^{NN}, s_{n+2l}^{NN}, \ldots, s_{n+(d-1)l}^{NN}).$$

If there is a large amount of data, the distance between them is relatively small.
Expanding the state space to $\Re^{d+1}$ yields the same points,

$$\hat{\boldsymbol{y}}_n = (s_n, s_{n+l}, s_{n+2l}, \ldots, s_{n+((d+1)-1)l})$$

and

$$\hat{\boldsymbol{y}}_n^{NN} = (s_n^{NN}, s_{n+l}^{NN}, s_{n+2l}^{NN}, \ldots, s_{n+((d+1)-1)l}^{NN}),$$

in a *d+1* dimensional space where the distance between them may or may not still be small. If the points where true neighbours they should separate relatively slow with successive increases in embedding dimension, and vice versa.

The increase in distance between the points is given only by the difference between the last components:

$$\| \hat{y}_n - \hat{y}_n^{NN} \|^2 - \| y_n - y_n^{NN} \|^2 = (s_{n+((d+1)-1)l} - s_{n+((d+1)-1)l}^{NN})^2 . \qquad (2.13)$$

The normalized increase in distance between these two points is calculated and the points are considered to be false nearest neighbours if

$$\frac{| s_{n+((d+1)-1)l} - s_{n+((d+1)-1)l}^{NN} |}{\| y_n - y_n^{NN} \|} \geq R_T . \qquad (2.14)$$

A suitable value for R_T depends on the spatial distribution of the embedded data. If R_T is too large, some false nearest neighbours will be missed, and if R_T is too small, some true neighbours will be counted as false. It is therefore important to choose a value for R_T that is suitable for the spatial distribution of the data under consideration. Typical values are $10 \leq R_T \leq 30$ (Small, 1998).

The dimension that corresponds to zero false nearest neighbours (or close to zero) is considered to be the minimum embedding dimension of the data; and is less than or equal to the sufficient dimension (remember that $d_e \geq 2d_0(A)$) found from the embedding theorem.

C. Embedding by Singular Spectrum Analysis (SSA)

An alternative approach to the method of delays (MOD) embedding is the use of *singular spectrum analysis* (SSA). Both methods are theoretically equivalent, but may differ in practice with limited amounts of possibly noisy data. Singular spectrum analysis (SSA) is based upon the theory of singular value decomposition and does not rely on the explicit calculation of embedding delay. Since the calculation of embedding delay is suspect in the presence of noise, the implicit calculation thereof makes SSA a particular attractive option (Barnard & Aldrich, 2000).

The first step in the SSA method of reconstruction is to derive a large *p*-dimensional state vector, $\Re^p$, from the scalar observations. This is achieved by embedding the

observations using a time lag of one, and an embedding dimension that is equal (at least) to the point of linear decorrelation of the observed time series. This point is the smaller of either the first minimum or the first zero crossing of the linear autocorrelation function. The value of the preliminary embedding dimension is the same as the delay value (l) used in a conventional, autocorrelation function determined, MOD reconstruction. Applying these parameters (l=1 and d=l) to the generalized reconstructed state space vector (***Equation 2.8***),

$$\boldsymbol{y_n} = (s_n, s_{n+l}, s_{n+2l}, \ldots, s_{n+(d-1)l}),$$

results in

$$\boldsymbol{y_{n,p}} = (s_n, s_{n+1}, s_{n+2}, \ldots, s_{n+l-1}) \tag{2.15}$$

as the preliminary, p(=l)-dimensional, reconstruction. The covariance matrix of the embedding is calculated, and the eigenvalues of the covariance matrix are determined and ordered to decreasing rank. This is also known as Singular Value Decomposition (SVD). The final d-dimensional state vector is a projection onto the first d principle components defined by the data in $\Re^p$, that is

$$\boldsymbol{y}_{n,d} = \boldsymbol{P} \times \boldsymbol{y}_{n,p} \tag{2.16}$$

where $\boldsymbol{y}_{n,d}$ = final d-dimensional reconstruction

$\boldsymbol{P}$ = first chosen principle components

$\boldsymbol{y}_{n,p}$ = preliminary p (=l)-dimensional reconstruction

The difference between SSA and MOD is that in MOD the d-coordinates are samples separated by a fixed delay (l) and cover a specific time window length, $l_w = (d-1)l$; while in SSA all the available samples in l_w are initially used and then further processed with SVD so that the final d-coordinates are linear combinations of these measurements. The free parameter, d, in SSA is not critical at all and any choice over a lower limit would give essentially the same reconstruction, because the additional coordinates correspond to less significant singular values and give negligible variance – assuming the time window length is sufficiently large (Kugiumtzis & Christophersen, 1997). To keep computational cost low it is sensible to choose smaller values for d, but

the reconstructed space, $\Re^d$, should still explain at least 95% of the variance of the original data.

The benefits of singular spectrum analysis are (Barnard & Aldrich, 2000):

a) Robustness against noise in the observed data
b) Simplicity of the calculations
c) Low numerical cost of the calculations
d) Significant reduction in measurement and dynamic noise

2.5 Dimension Estimates: Invariants of the Dynamics

Attractor dimension has been the most intensely studied invariant quantity for dynamical systems (Abarbanel, 1993). Generally, people think of real world objects as being one, two or three-dimensional. There, however, exist complex mathematical objects, known as fractals, which have non-integer dimensions. Attractors associated with regular (or integrable) systems always have an integer dimension, whereas attractors associated with chaotic dynamics usually have a fractal dimension. Evidence of fractal dimensions in experimental time series data demonstrates the existence of chaos in the real world, and this is what stimulates research in the area.

Fractal dimensions are characteristic of the geometry of the attractor and relate to the way points on the attractor are distributed in *d*-dimensional space. It is invariant under the evolution operator of the system and is therefore independent of changes in the initial conditions of the attractor orbit. It is also independent of the coordinate system in which the attractor is observed, which means that it can be determined from the *reconstructed state space*. In other words, the fractal dimension of a dynamic system can be estimated from experimental data, which makes it an extremely valuable statistical quantity for classifying dynamical systems from experimental data.

2.5.1 The Box-counting Dimension

The simplest concept of dimension is the number of coordinates needed to specify a state. This idea can be taken a step further by using the geometrically related concept of dimension, whereby volume scale as a function of a characteristic length (ε) parameter:

$$V(\varepsilon) \propto \varepsilon^d \tag{2.17}$$

Here (V) is a measure of the volume, which can be length (one-dimensional), area (two-dimensional), volume (three-dimensional) or hyper-volume (multi-dimensional), depending on the geometry. The length scale is denoted by ε (e.g. the length of a cube's side or the radius of a sphere) and the dimension of the object by *d*. If assumed that the relation in ***Equation (2.17)*** holds true for a general fractal, a universal relation to define dimension can be obtained:

$$d = \frac{\log V(\varepsilon)}{\log \varepsilon} \tag{2.18}$$

First attempts to estimate dimension involved calculating the volume of the attractor. This is done by partitioning the *d*-dimensional attractor in phase space with a ε-sized grid and counting the amount of partitions that contain at least one data point. The amount of partition elements that conform to this criterion is used as a measure of the volume at that resolution ε. The procedure is repeated, each time with a smaller grid size, until ε has spanned a large enough range (theoretically $\varepsilon \to 0$). This approach to estimate the dimension is known as the *box-counting* algorithm and the calculated dimension is called the *box-counting dimension* (d_0).

The largest possible value of d_0 that can be obtained through these calculations is the dimension used to do the initial embedding (d_e). If the embedding is two dimensional, but the attractor has not unfolded completely, one will get a (incorrect) value of $d_0 = 2$. Embedding the data in increasingly higher dimensions will yield $d_0 = d_e$, until the attractor's geometrical structure is fully unfolded. At this point higher dimensional embeddings will not increase the *box-counting dimension* (d_0) significantly. This saturated fractal dimension is the proper value for d_0.

In ***Equation (2.16)*** the volume is related to the amount of points in the space. Theiler (1987) proposed a more general form of ***Equation (2.16):***

$$bulk = size^{\dim ension} \tag{2.19}$$

which results in

$$d = \frac{\log(bulk)}{\log(size)} = \frac{\log(bulk)}{\log(\varepsilon)} \tag{2.20}$$

The past few years researchers has spend quite some time developing techniques to improve the measuring of "bulk", resulting in various dimension estimators. The generalized dimension (d_q) is defined by

$$d_q = \lim_{size \to 0} \frac{\log(bulk)}{\log(size)} = \lim_{\varepsilon \to 0} \frac{1}{q-1} \frac{\log \sum \rho^q}{\log \varepsilon} \tag{2.21}$$

where ρ is the natural density of points in the phase space. This definition of d_q provides a whole spectrum of invariant quantities for $-\infty < q < \infty$. A more detailed explanation of how d_q is derived from basic principles can be found in the article by Abarbanel (1993). Letting $q = 0$ in ***Equation (2.21)*** results in

$$d_0 = \frac{-\log N_0}{\log \varepsilon}, \tag{2.22}$$

which is simply the definition of the *box-counting dimension* as previously discussed.

2.5.2 The Information Dimension

The *information dimension (d_1)* is another frequently cited dimension estimate. As with the *box-counting dimension,* the attracter is partitioned by a ε-sized grid in the phase space. Instead of simply counting the number of partitions containing data points, the "volume" of the attractor contained within each partition is calculated. This measure seeks to account for differences in the distribution density of points covering the attractor.

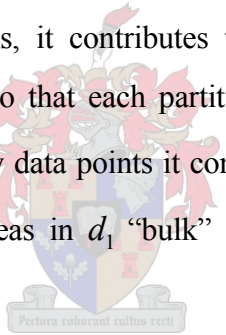
The information dimension (d_1) is defined by letting $q = 1$ in the generalized dimension (d_q) (***Equation 2.21).*** Using L'Hopital's rule:

$$d_1 = \lim_{\varepsilon \to 0} \frac{\sum_i p_i \log p_i}{\log \varepsilon} \tag{2.23}$$

where p_i is the probability of data points from the attractor occurring within the i^{th} element of the partition:

$$p(x_i) = \frac{n(x_i)}{\sum_i n(x_i)}$$

The difference between the measures d_0 and d_1 is in the way the "bulk" (***Equation 2.19)*** is calculated. The d_0 statistic measures "bulk" by counting the number of nonempty partitions in the phase space. This means that regardless of how many data points a ε-sized partition contains, it contributes the same amount to the "bulk". Contrary to this, d_1 is calculated so that each partition element's contribution to the "bulk" is proportional to how many data points it contains. "Bulk" in d_0 is simply the volume the attractor covers, whereas in d_1 "bulk" is equivalent to the mass of the attractor.



2.5.3 The Correlation Dimension

The calculation of the *box-counting dimension* (d_0) and *information dimension* (d_1), in practice, usually requires a prohibitive amount of computation time. This is why the most commonly used dimension estimate for attractors is the *correlation dimension* (d_c). Calculation of the *correlation dimension* (d_c) is computationally efficient and relatively fast when implemented as an algorithm for dimension estimation.

Consider the case where $q = 2$ in the generalized dimension (***Equation 2.21***):

$$d_2 = \lim_{\varepsilon \to o} \frac{-\log \sum_i p_i^2}{\log \varepsilon} \tag{2.24}$$

Here d_2 refers to the measure known as the *correlation dimension* (d_c). The numerator in ***Equation (2.24)*** constitutes a two-point correlation function which measures the probability of finding a pair of random data points within a specific partition element (Compare this to the significance of the numerator in d_1: ***Equation 2.23***) . The numerical value of $\sum_i p_i^2$ can be estimated by counting how many pairs of points have a separation distance less than some value ε.

In 1983 Grassberger and Procaccia suggested a simple and computationally efficient way of estimating the $\sum_i p_i^2$ - term in the *correlation dimension* equation. Let $\{\boldsymbol{y}_i\}_{n=1}^N$ be an embedding of a time series in $\Re^{d_e}$. The correlation function, $C_N(\varepsilon)$, is defined by:

$$C_N(\varepsilon) = \binom{N}{2}^{-1} \sum_{0 \le i \le j \le N} I\left(\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\| < \varepsilon\right) \tag{2.25}$$

Here $I(X)$ is a Heavyside function whose value is one (1) if condition X is satisfied and zero (0) otherwise. The $\|\cdot\|$-term is the distance function in $\Re^{d_e}$. The points, $\boldsymbol{y}_i$, are points on the reference trajectory and $\boldsymbol{y}_j$ are other points on the attractor in the vicinity of $\boldsymbol{y}_i$. If the distance between point $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is within ε, $I(X)$ takes a value of one; and if the distance between the points is outside ε it takes a value of zero. This means that the sum $\sum_i I\left(\left\|y^i - y^j\right\| < \varepsilon\right)$ is the number of points within a distance ε of point $\boldsymbol{y}_i$.

The calculation and interpretation of the correlation dimension estimates forms an integral part of this thesis, so it is worth spending time understanding the physical meaning of ***Equation (2.25).*** The idea is illustrated by ***Figure 2.9*** where a hypersphere of radius ε is centred on one of the points defining the trajectory of the attractor.

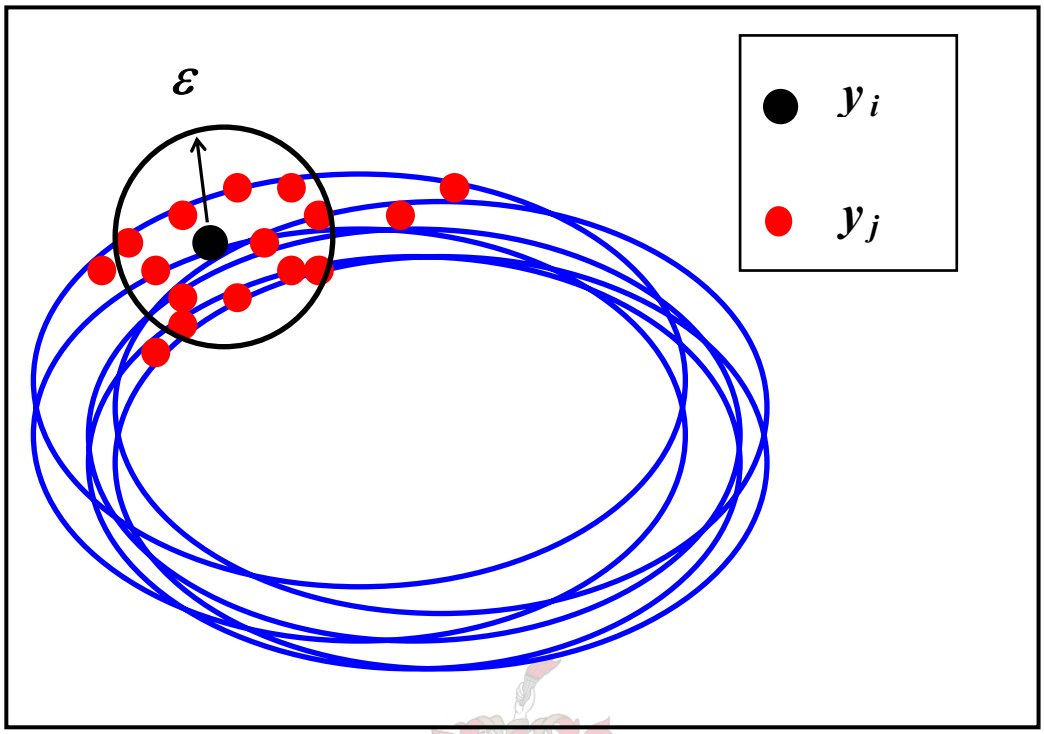


Figure 2.9: *Probing hypersphere on the attractor*

The number of other points on the attractor within the hypersphere is counted. The hypersphere is then moved from point to point along the trajectory and each time the number of points within is counted. The cumulative sum of points is then divided by $N(N-1)$ to give the correlation sum (C_N) for a hypersphere of a particular size. For each value of the hypersphere size (ε) there will be a different value for the correlation sum (C_N). The maximum value of C_N is unity and this happens when the radius of the probing hypersphere is large enough to include the whole attractor, i.e. all the points of the attracter are counted. The minimum value of C_N is $2/[N(N-1)]$ and occurs when only the closest two points on the attractor are counted. One should notice that the correlation sum counts the closest near neighbour twice as the probing hypersphere visits both points on its journey around the attractor. This is the reason for the 2 in the numerator (Addison, 1997)

A. The Grassberger-Procaccia Algorithm

The method traditionally most often employed to estimate the correlation dimension is the Grassberger-Procaccia algorithm (Grassberger & Procaccia, 1983). They used the assumption that the correlation sum (C_N) scaled with the hypersphere radius (ε) according to a power law in the form

$$C_N \propto \varepsilon^{d_c}, \tag{2.26}$$

where d_c is the correlation dimension of the attractor.

In this method, the correlation sum (C_N) is calculated for different sizes of the hypersphere radius (ε). A graph is constructed that plots $\log C_N(\varepsilon)$ versus $\log \varepsilon$. The gradient of this graph in the limit as $\varepsilon \to 0$ should approach the correlation dimension. The problem, however, is that the graph will jump about irregularly for small values of ε when using a finite amount of data. This can be overcome by looking at the graph when using moderately small ε-values. A typical curve (***Figure 2.10***) from this plot will have three distinct features:

1) For large hypersphere sizes (ε) the graph will flatten. The flattening happens because the attractor has a finite size.

2) At moderate ε-sizes the graph has a "scaling region" which is approximately linear.

3) For small ε-sizes the graph jumps about irregularly. The irregularity is a result of having only a finite amount of data, so that the contribution to the correlation function by a specific pair of points is seen as a jump in the graph.

The extremes of the graph coincide with very large and very small scales that tend to be nonlinear. The correlation dimension is calculated from the slope of the "linear" part of the graph.

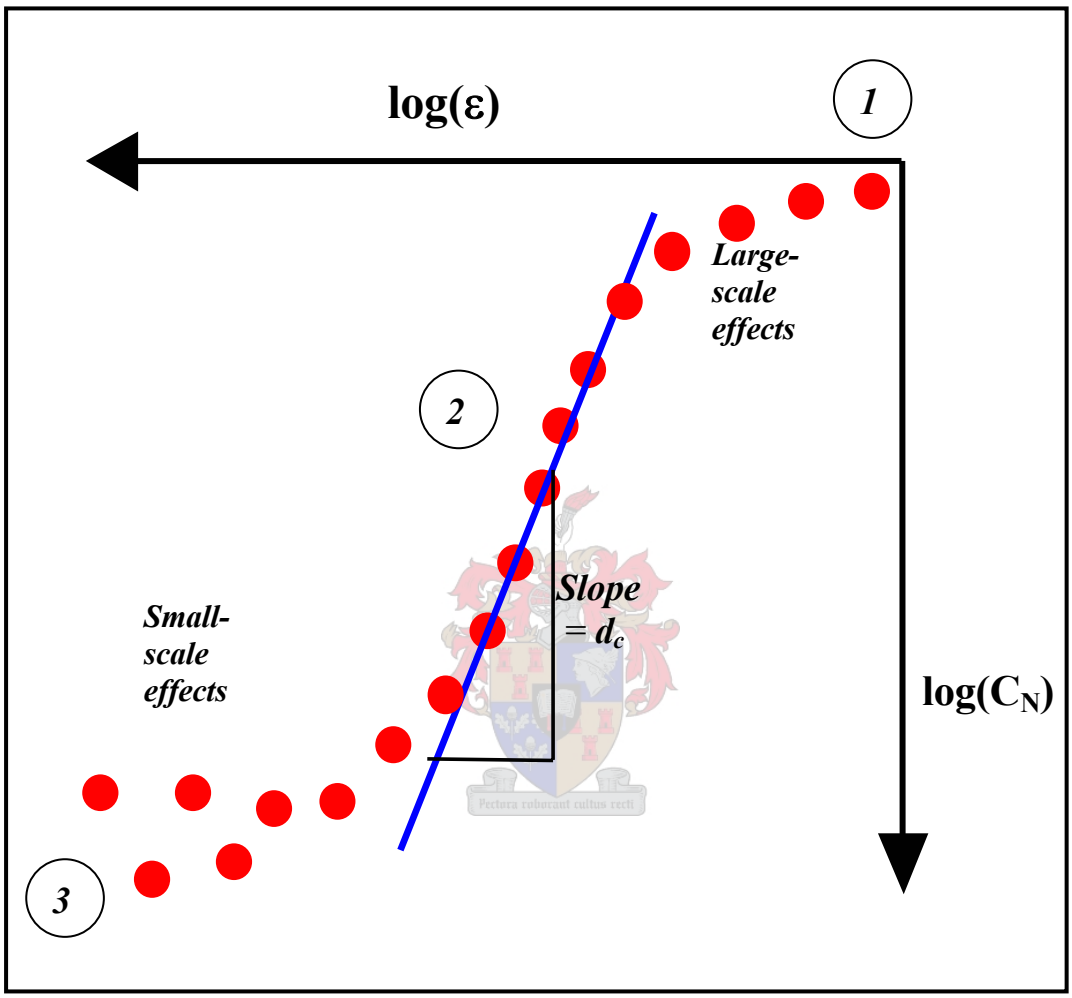


Figure 2.10: *The log(ε)-log(C_N) plot for determining the correlation dimension via the Grassberger-Procaccia algorithm.*

B. Judd's Algorithm

Conventional implementations of the Grassberger-Procaccia method assume that most of the information about the dimension is contained in the linear (scaling region) of the log(ε)-log(C_N) plot and proceed by fitting a straight line to the scaling region. As Judd (1992) points out there are deficiencies in the Grassberger-Procaccia algorithm:

1) From a sample trajectory of length n there are $n(n-1)/2$ interpoint distances, but these distances are not all *independent*. The triangle inequality states that the distance between two points is no more than the sum of the distances from a third point. This problem is inherent to all methods that are based on calculations of interpoint distances.

2) The smoothness in the correlation function is misleading because it contains a lot of statistically correlated information. This is because the value of $C_N(\varepsilon)$ is based on the number of points in the ε-sized hypersphere, which becomes more statistically correlated as ε is increased. Any fit of this function should take this into account by giving the larger values of ε less weight.

3) It is possible that the scaling (linear) region may not reflect information about the dimension of an attractor. According to Judd (1992) examples can be constructed were the scaling region reflects only large-scale properties of an attractor, and not the dimension. The scaling region is also a subjective, but critical choice, and sometimes a small change in the scaling region significantly changes the correlation dimension estimate. It often happens that the analyst chooses a scaling region that is not straight, but curved. When using a certain scaling region, information about small distances between points (which is still information about scaling) is ignored. There is no basis on which this information can be thrown away.

He also states in a later paper (Judd, 1994) that linear correlation in the data set misleads the algorithm to falsely show convergence to some low dimension, which could then be misinterpreted for inherent low-dimensional dynamics.

Judd (1992) proposes a different algorithm for calculating the correlation dimension. His algorithm is a modification of the Grassberger-Procaccia algorithm that, in contrast to conventional implementations, considers the contribution of the interpoint distances directly and not the correlation function. The significance of this is that problems (2) and (3) are completely avoided. He demonstrates that the following equation is a better description of the correlation function, valid for $\varepsilon < \varepsilon_0$:

$$C_N(\varepsilon) \approx \varepsilon^{d_c} q(\varepsilon) \tag{2.27}$$

where $q(\varepsilon)$ is a polynomial of order *t*, the topological dimension of the set. The topological dimension is the smallest dimension of embedding space in which the attractor fully unfolds. The correlation dimension (d_c) is a function of ε_0 and written as $d_c(\varepsilon_0)$.

Where the Grassberger-Procaccia method assumes a linear scaling region by fitting $C_N \propto \varepsilon^{d_c}$, the new method of Judd fits $C_N(\varepsilon) \approx \varepsilon^{d_c} q(\varepsilon)$ which allows for the presence of an additional polynomial term that takes into account variations of the slope within and outside of the scaling region. Judd's method dispenses with the need for a scaling region and substitutes a single scale parameter, ε_0. This has an interesting benefit, which is especially useful for the purpose of detecting dynamic changes (which will become evident in ***Chapter 3***). Many natural objects do not have a constant dimension at all length scales. Consider a large river stone as an example. If one examines the surface at its largest length scale, it is almost two-dimensional. However, at smaller length scales (when the stone is examined closely and in more detail) one can discern the details of grains, which add to the complexity and increase the dimension at smaller scales. This is the same with attractors reconstructed from time series data. Have a look at the reconstructed attractor in ***Figure 2.11.*** At its largest length scale (***Figure 2.11(a)***) the attractor seems to be low dimensional. However, when zooming in on a smaller part of the attractor (***Figure 2.11(b)***), i.e. examining the smaller length scales, it is obvious that the complexity and therefore the dimension of the attractor increases.

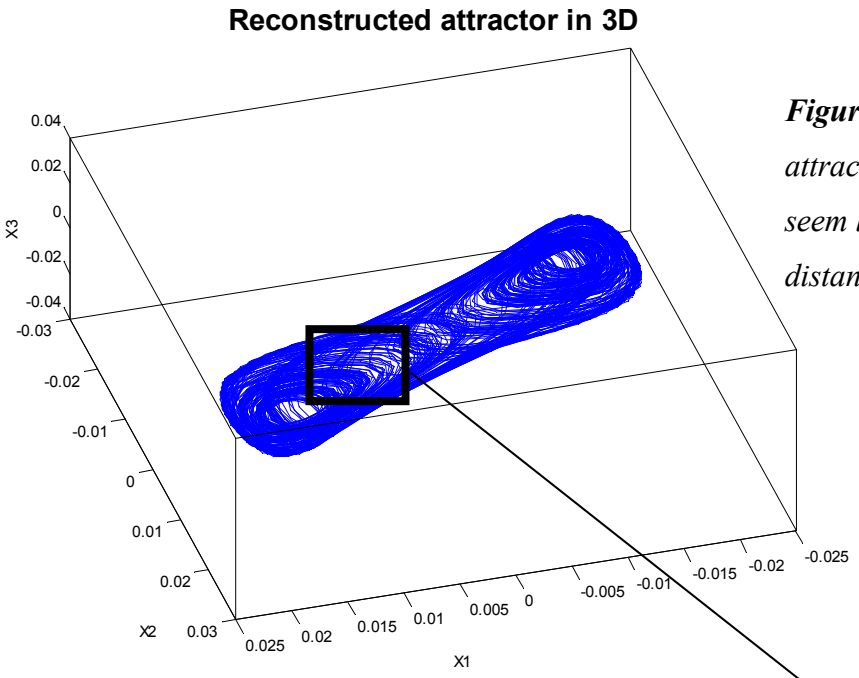


Figure 2.11(a): *A reconstructed attractor from Chua's circuit that seem low dimensional from a distance.*

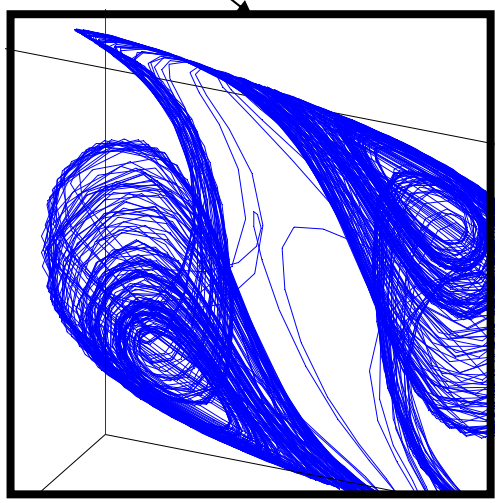


Figure 2.11(b): *Zooming in on part of the attractor reveals the high dimensional nature of the object*

It is therefore natural to consider the dimension (in this instance the correlation dimension) as a function of ε. ***Figure 2.12*** illustrates how the correlation dimension for the attractor in ***Figure 2.11*** change with different length scales. As expected the correlation dimension (d_c) has smaller values at larger length scales, and vice versa.

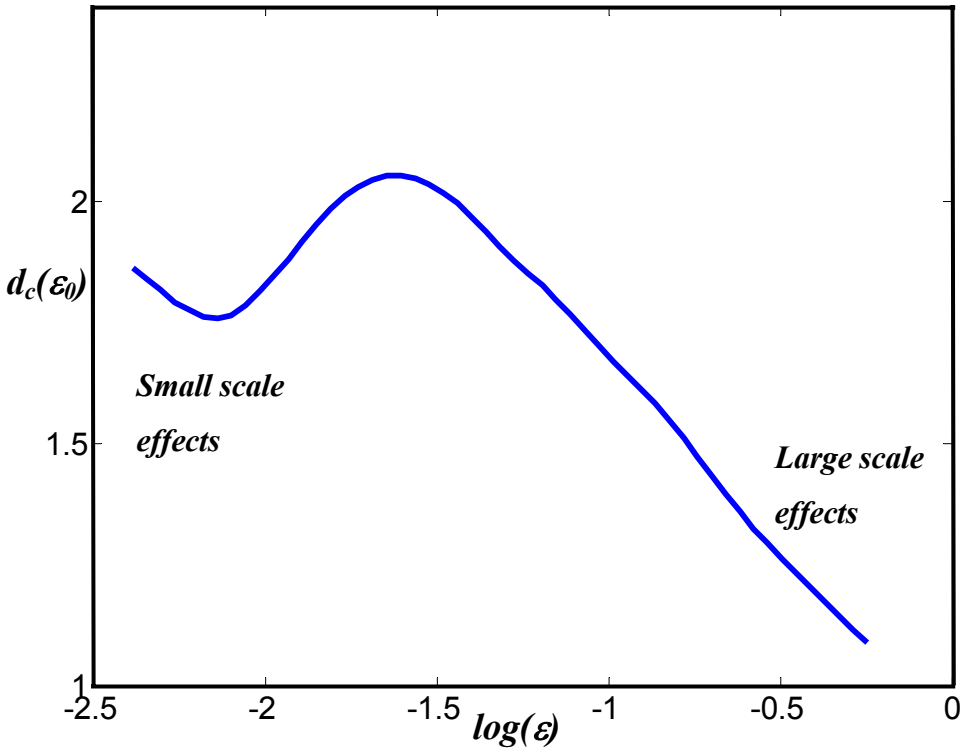


Figure 2.12: *A typical graph illustrating Judd's method where d_c is a function of ε.*

By allowing the correlation dimension to be a function of length scale, estimates that are both more accurate and informative can be produced. Some of the approximation necessary to define correlation dimension as a single number can be avoided when using Judd's method, but most important, more detailed information on the topology of the attractor can be extracted.

2.6 Surrogate Data Analysis

Data from nonlinear deterministic processes require more complex and time-consuming analysis techniques (i.e. state space reconstruction) that are much less understood than the classical methods used for linear stochastic process data. Before blindly applying these advanced data analysis and modelling techniques, it is important to check whether the data exhibits such nonlinear behaviour that would deserve more advanced treatment.

It would be particularly useful to have answers to the following questions (Kantz & Schreiber, 1997):

- Is the apparent structure in the data most likely owing to nonlinearity or rather due to linear correlations?

- Is the irregularity of the data most likely due to nonlinear determinism or rather due to random inputs to the system or fluctuations in the parameters?

The analyst should be aware that linear stochastic processes can also create very complicated looking signals and that not all irregularities in a data set are due to nonlinear dynamics within the system.

To confirm this statement, consider the following simple linear stochastic process:

$$x_n = 0.99x_{n-1} + \eta_n \tag{2.28}$$

Here η_n are independent Gaussian random numbers. Successive values in this linear time series are strongly correlated and already yield some structures. ***Figure 2.13*** illustrates the irregular behaviour of this process.

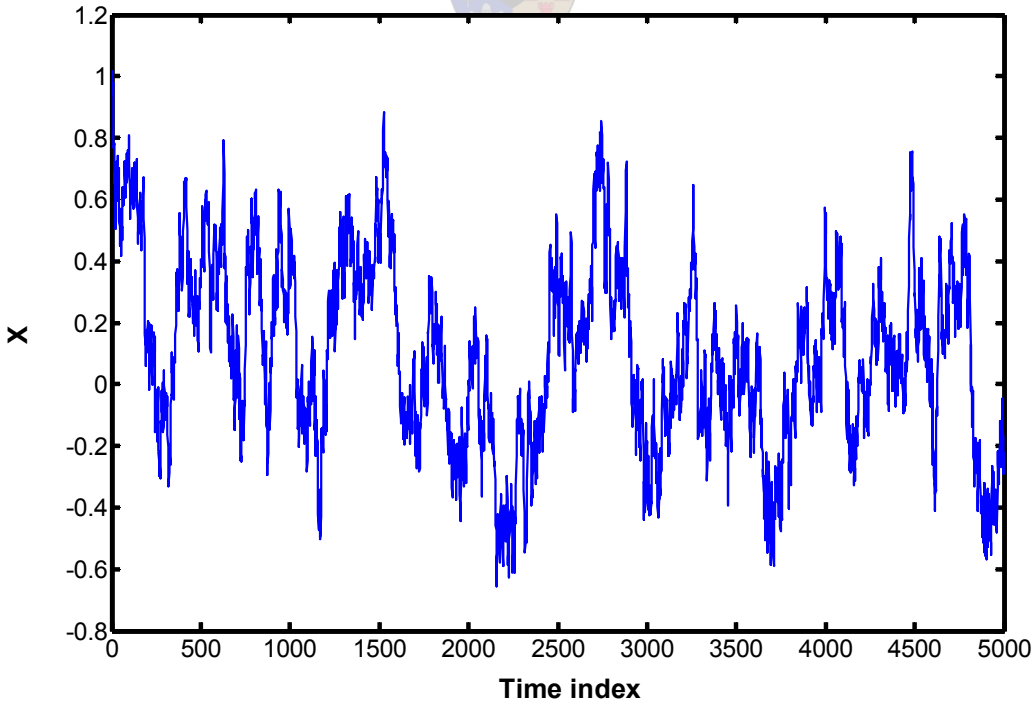


Figure 2.13: *Irregular output from a simple linear process*

The output can be made even more complex by observing the values from the process through a nonlinear measurement function:

$$s_n = x_n^3 \tag{2.29}$$

Note that although the measurement function is nonlinear, there are no nonlinear dynamics in this process. All the dynamics is contained in the linear AR(1)[9] part and the nonlinearity is purely static. ***Figure 2.14*** shows the spiky irregular output from such a linear system.

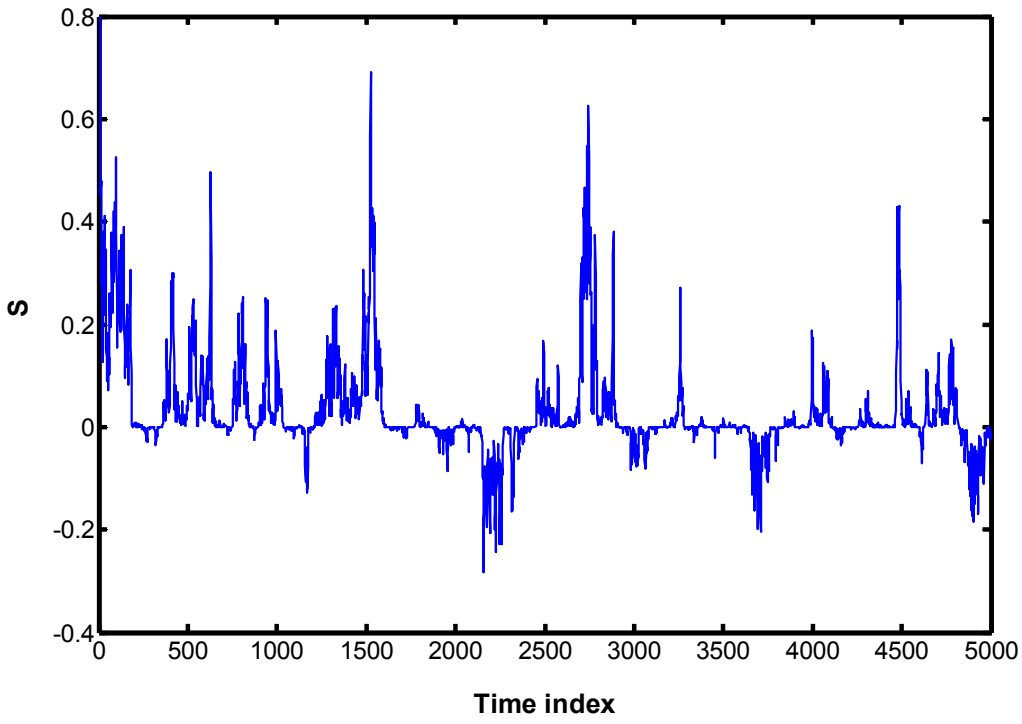


Figure 2.14: *Irregular output from linear process observed through a nonlinear measurement function*

A time series can be anything between purely random and strictly deterministic. When looking at the previous examples it is understandable that linear structures can easily be mistaken for (nonlinear) determinism. It is therefore important to have a reliable test to classify the given time series with confidence.

[9] Autoregressive process of order one.

The test most successfully applied by researchers and analysts, makes use of the method of *surrogate data* (Theiler et al., 1992). This test involves a null hypothesis against which the data are tested, using a discriminating statistic. The data under investigation are first assumed to belong to a specific class of dynamic processes, e.g. linear stochastic. Surrogate data sets are generated from the original data, using this assumption. An appropriate statistic then is calculated for both the original data and the surrogate data sets. If the resulting statistic is significantly different for the surrogate data than for the original data, the null hypothesis is rejected. This means that the process (linear stochastic in this instance) that generated the surrogate data sets is not from the same class than that of the original data. By repeating this procedure, and through trail-and-error elimination, it is possible to get a good idea of the characteristics of the original process (Barnard et al., 2000). One should always progress from simple and specific assumptions to broader and more sophisticated models.

More formally, the test can be explained as follow. Let ψ be a specific hypothesis and F_ψ be the set of all process systems consistent with that hypothesis. The time series (s_n) under investigation, which consists of *n* scalar measurements, can be expressed as $s \in \Re^n$. Let $T : \Re^n \to U$ be a statistic which will be used to test the hypothesis that s was generated by some process $F \in F_\psi$. Generally, $\boldsymbol{U}$ will be $\Re$, and it will be possible to discriminate between the original data (s) and the surrogates (s_i) consistent with the hypothesis given by the approximate probability density $p_{T,F}(t)$, i.e. the probability density of *T* given *F* (Small & Judd, 1998).

2.6.1 Hypothesis Testing

It is important to realise that it is **only** possible to **reject** a hypothesis; it is not possible show that it is correct. Data can be consistent with a hypothesis even if the hypothesis is **not** correct. However, if data are inconsistent with the hypothesis, the data show that the hypothesis is not correct. So, although data can definitely reject a hypothesis, it can only suggest in an informal way that a hypothesis is correct. One should keep in mind that a hypothesis can contain many hidden assumptions and if a hypothesis is rejected, one may actually be rejecting these hidden assumptions.

Classes of hypothesis

Different types of surrogate data are generated to test membership of specific dynamical system classes, referred to as hypotheses. Three commonly used classes of hypothesis are discussed here. They are equivalent to the following assumptions:

1) The data are identically, independently distributed noise (***type 0***)

2) The data are linearly filtered noise (***type 1***)

3) The data are a monotonic non-linear transformation of linearly filtered noise (***type 2***)

To test if the data are consistent with a particular hypothesis, a model is build that is consistent with that hypotheses and has the same properties as the original data. Surrogate data are then generated from the model:

1) ***Type 0*** – Shuffles (randomises) the data

2) ***Type 1*** – Randomizes the phases of the Fourier transform of the data

3) ***Type 2*** – Applies a phase randomising procedure to amplitude adjusted Gaussian noise. The procedure for generating ***type 2*** surrogates can be seen as a combination of generating ***type 0*** and ***type 1*** surrogates:

   i. A normally distributed data set z is generated from the original data set s, and reordered so that z has the same rank distribution as s.

   ii. A ***type 1*** surrogate set z_i is generated from z by randomising the phases of the Fourier transform of z.

   iii. Finally, the original data s is reordered to create a surrogate set s_i which has the same rank distribution as z_i.

These ***type 2*** surrogates are also referred to as amplitude adjusted Fourier transformed (AAFT) surrogates.

The original data are now compared to the different surrogate data sets to if it is typical under the hypothesis. The surrogates generated by the algorithm 0, 1, or 2 uses a linear model. Each of these surrogate tests addresses a hypothesis that the data are either linear, or some (linear or monotonic nonlinear) transformation of a linear process.

2.6.2 The Test Statistic

To compare the original data to its surrogate data, a appropriate test statistic must be selected. Such a statistic should measure a non-trivial invariant of a dynamical system that is independent of the way surrogates are generated. Theiler and Prichard (1996) suggested that there are two, fundamentally different types of test statistics: pivotal and non-pivotal. A statistic (T) is considered pivotal if the probability distribution of T ($p_{T,F}$) is the same for all processes (F) consistent with the hypothesis (ψ). The probability distribution ($p_{T,F}$) is therefore invariant for all $F \in F_\psi$. One should also make the distinction between a *simple* hypothesis and a *composit*e hypothesis. If the set of processes consistent with the hypothesis (F_ψ) is singleton, the hypothesis is *simple*. Otherwise, the hypothesis is composite and the problem is twofold: generating surrogates consistent with a particular process (F), as well as estimating $F \in F_\psi$.

Theiler and Prichard (1996) argues that F has to be specified when the hypothesis is composite, unless the test statistic (T) is pivotal – in which case the distribution of T is the same for all $F \in F_\psi$. They suggest that a constrained realization scheme should be employed when non-pivotal statistics are to be applied to composite hypotheses. This implies one must ensure that the surrogates are realizations of a system consistent with the hypothesis that gives identical estimates of the parameters of that system when compared to the estimates of those parameters obtained from the original data, instead of generating surrogates that are typical realizations of the model. So if $\hat{F} \in F_\psi$ is the process estimated from the data (s) and s_i is a surrogate data set generated from $F_i \in F_\psi$, then the process ($\hat{F}_i \in F_\psi$) estimated from the surrogate data set (s_i) must be the same as $\hat{F}$.

For example, let ψ be the hypothesis that the original data (s) is generated by linearly filtered i.i.d.[10] noise. *Non-constrained* surrogate data (s_i') can be generated from a Monte Carlo simulation based on the best linear model estimated from s. The *non-constrained* surrogate data (s_i') can be constrained by shuffling the phases of the Fourier transform. This will produce a set of random data (s_i'') with the same power spectra and autocorrelation as the original data (s). Autocorrelation, nonlinear prediction error or rank distribution statistics would all be non-pivotal test statistics, as the distributions of these statistics would all depend on the form of the noise source and type of the linear filter. However, the *correlation dimension* (as well as other measures derived from dynamical system theory that are invariant under diffeomorphisms) would be pivotal statistics. The probability distributions of these quantities would be the same for all processes, despite the source of noise or the estimated model.

The use of the correlation dimension (d_c) as pivotal statistic of choice, has gained a great deal of favour over the years. One of the reasons being that there exists a very powerful and reliable method of estimating d_c (as described in the previous sections); which gives a quick, effective and informative method for classifying time series data (Small & Judd, 1998).

The power of *correlation dimension* (d_c) as a test statistic can be illustrated with an example. Consider the two data sets in ***Figure 2.15***. It is impossible to tell with certainty whether they exhibit nonlinear deterministic behaviour and if both are simply stochastic. Fifteen surrogate data sets (between 15 and 30 sets are needed for statistical significance) are created from each of the two data sets. To compare the surrogate and actual data sets with each other, both are embedded into a higher dimensional space than was calculated from the original time series data. The assumption is that a lower dimensional attractor from a deterministic system will occupy a subspace of the higher dimensional space and should not extent into new coordinates. Usually an embedding dimension of 10 is enough to ensure that the attractor has fully unfolded. The original data (s) is then compared with the surrogate data sets (s_i) by calculating the correlation dimension (d_c) from these reconstructions.

[10] Independently and identically distributed

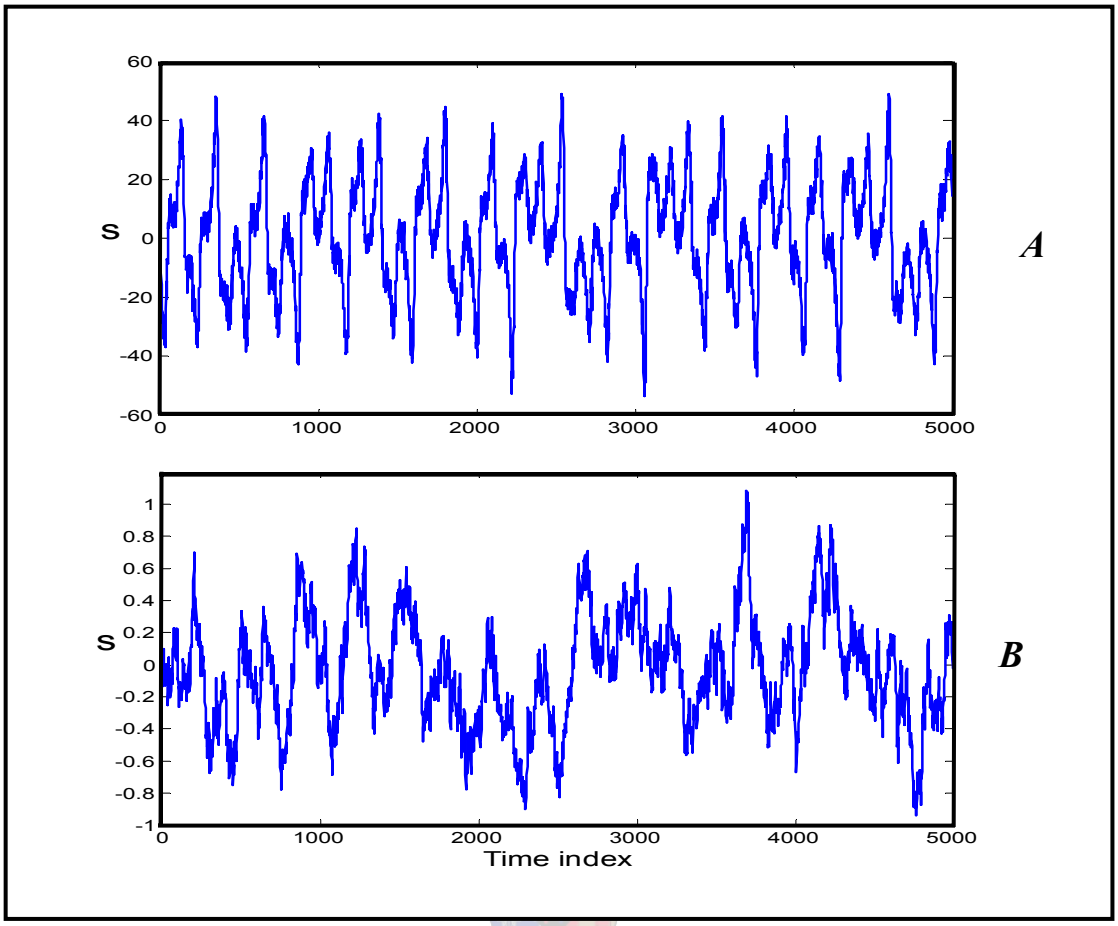


***Figure 2.15**: Two data sets that need to be classified.*

From ***Figure 2.16*** it can be seen that there is a distinct separation between the cluster of surrogates and the curve of the original data. The ***type 2*** hypothesis can therefore be rejected with confidence, which indicates that there is a strong possibility for determinism in the data. The next graph, ***Figure 2.17***, shows quite the opposite. The curve of the original data is part of the cluster of surrogate data. It is therefore clear that this data came from a linear stochastic process

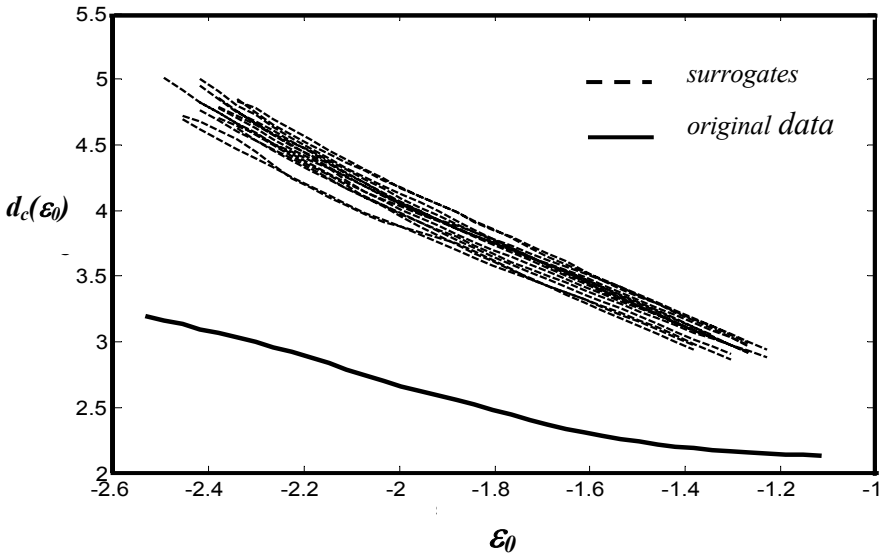


Figure 2.16: *Correlation dimension plot of surrogates and original data for data set A.*

.

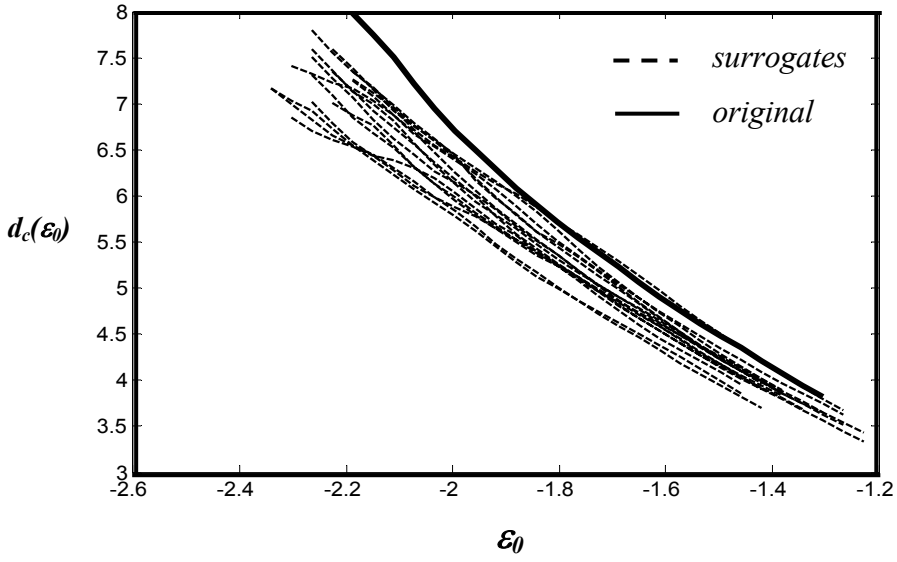


Figure 2.17*: Correlation dimension plot of surrogates and original data for data set B.*

Dataset ***A*** comes from a nonlinear deterministic system, namely the Lorenz system (Lorenz, 1963). The dataset ***B*** was generated by ***Equation (2.28),*** which generated linear stochastic data. The results from these two examples show that surrogate data analysis, with correlation dimension as test statistic, is a powerful data classification technique.

A more detailed discussion on surrogate data analysis can be found in the articles by Schreiber & Schmitz (2000) and Kaplan (1999).

3 DETECTING DYNAMIC CHANGE

A problem frequently encountered when analyzing chemical process systems, is appropriate characterization of changes in the system dynamics. Since the discovery of time delay embedding for state space reconstruction, a considerable amount of work has been devoted to the development of techniques to extract information in observed time series data through analysis of the geometrical structure of the attractor underlying the time series. Underlying nearly all of these techniques (of which some are discussed in ***CHAPTER 2***) is the assumption that the dynamical process under question is *stationary*. This means that the dynamical process, and therefore the geometrical attractor, has not changed on long time scales – i.e. time scales in the order of the length of the data set. If the geometrical shape of the attractor does change during this time, denoting a *nonstationary system*, there are two possible reasons:

1) There may be significant behaviour on time scales longer than may be reliably resolved with the given data; meaning that the given data set (time series) is too short to capture the total dynamic behaviour of the process.

2) Process parameters, presumed to be fixed, have changed during the observation period.

In short, this implies that if there are deterministic rules governing the dynamics, these rules must not change during the time covered by the given time series data.

The first instance of *nonstationarity* can be illustrated using the Lorenz system (Lorenz, 1963) as an example. This three-dimensional system is well known for its nonlinear chaotic behaviour, and is described by the following coupled differential equations:

$$\frac{dX}{dt} = 10Y - X$$

$$\frac{dY}{dt} = 80X - Y - ZX$$

$$\frac{dZ}{dt} = XY - \tfrac{8}{3}Z \qquad (3.1)$$

Initially, the equations are solved by varying the time (t) from zero to 1.7; generating 5000 uniformly sampled data points. The resulting attractor, when plotting the variables X,Y and Z against each other, is shown in ***Figure 3.1***.

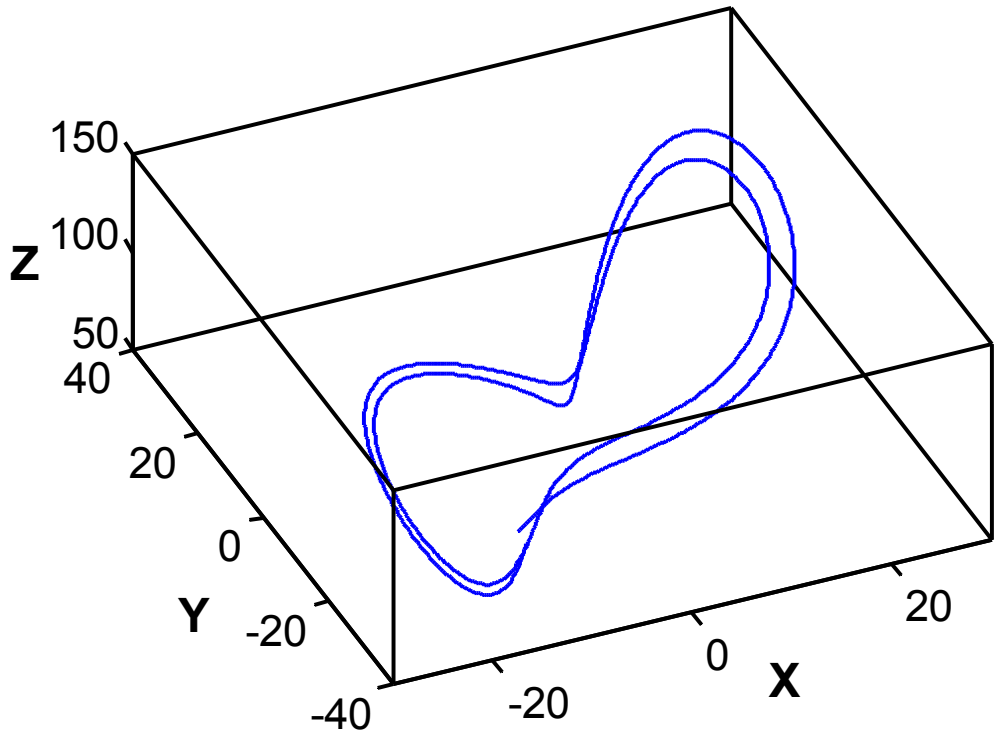


Figure 3.1: *Lorenz attractor for* $\boldsymbol{t} = 0:1.7$

A totally different attractor unfolds from the system when the number of time steps is increased to $t = 20$ (***Figure 3.2***). The resulting attractor is much fuller and display more dimensional detail.

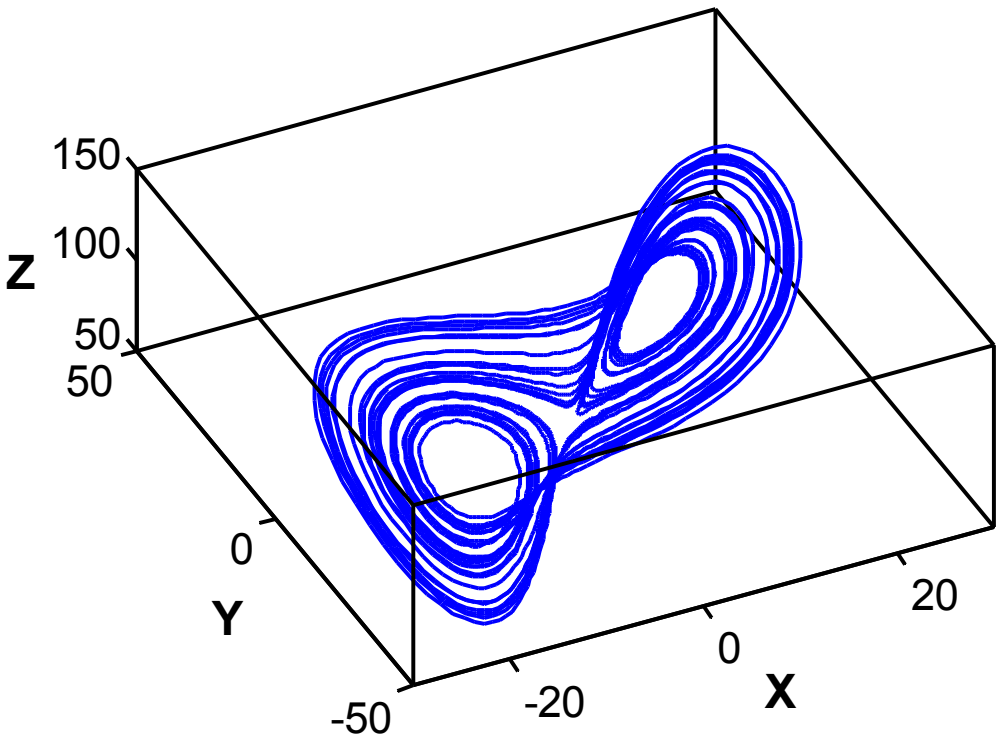


Figure 3.2: *Lorenz attractor for* $\boldsymbol{t} = 0:20$

It is obvious that trying to model the system using the first, shorter time series will be disastrous. The time series is clearly nonstationary, since it is not sufficiently long[1] to trace out a good approximation of the invariant measures.

The second instance of nonstationarity is more complicated. Changes in the process dynamics can be as result of either a slow drift in the system's parameters during the measurement period, or a near instantaneous change under the influence of the environment. Sometimes it is possible to handle a simple change of parameters once it is noticed. If the calibration of the measurement apparatus drifts, one can try to rescale the data continuously in order to keep the mean and variance constant. This is potentially dangerous, unless one is sure that it is the measurement scale, and not the process dynamics, that is drifting. In some cases where a strictly periodic modulation of

[1] ***Long*** in this context does not necessarily mean the number of data points, but rather the time-evolution.

a parameter is established, it can be treated as a dynamic variable rather than a parameter and does not necessarily destroy stationarity (Kantz & Schreiber, 1997).

More formally, a time series is called stationary if all transition probabilities from one state of the system to another are independent of *time* within the observation period, i.e. when calculated from the data. Detecting nonstationarity can be viewed as determining whether the dynamics underlying the system is autonomous or not. In the framework of deterministic systems, autonomous dynamical systems can be represented as being generated by time independent flow:

$$x_{n+1} = F(x_n) \tag{3.2}$$

Here x is a set of dynamic variables, and the reconstruction of a time series generated by such a system is sufficient to uniquely determine the flow that generated it. A nonautonomous dynamical system, however, is generated by time dependent flow:

$$x_{n+1} = F(x_n, t) \tag{3.3}$$

In this case the reconstruction of a time series is insufficient to determine the flow that generated it, and therefore the time series is considered nonstationary. For a closed, deterministic description of the system it is necessary to add time as a reconstruction variable. The problem is that such a reconstruction (with time as a variable) provides little useful information. With only one point in the phase space for each time, there is little predictive power for the outcome of a repeated experiment. That is unless the resultant time series is identical. For a variable to provide more useful information, its dynamics should be recurrent in the phase space.

Equation (3.3) can also be rewritten in the form:

$$x_{n+1} = G(x_n, y_n), \qquad y_{n+1} = H(y_n, t) \tag{3.4}$$

Here x carries information about the internal dynamics of the system, and y is the external driving force seeing as its dynamics is independent of x. If y is nonstationary, the entire process is nonstationary, and vice versa. Examples where y is stationary include cases where y is periodic, or is itself generated by another ergodic flow. One should keep in mind that ergodic systems are stationary, and that stationary systems are autonomous. Also, note that stationarity is a statistical concept, while

autonomy is a term that refers to the generating algorithm of the process (Manuca & Savit, 1996).

3.1 Measures for detecting nonstationarity

After explaining the stationarity problem in the former part of the chapter, the problem at hand is to come up with solutions on how to, for a given time series, detect nonstationarity. What makes this analysis so complicated is that stationarity is a property which can never be positively established, and because stationarity requirements differ depending on the application, the task is even more complex.

When testing for stationarity, the first solution one thinks of is to calculate a number of simple statistics, e.g. the mean and variance, from different parts of the time series and use a standard statistical hypothesis test based on their presumed equality. Conventional statistical process control (SPC) techniques, such as the standard Shewhart control charts, cumulative sum (CUSUM) control charts and various moving average (MA) control charts, are widely used to monitor the output of industrial processes (Basseville & Nikiforov, 1993). The main problem underlying these techniques is that they are designed to operate on *constant mean* processes, i.e. processes that are considered to be stationary within the linear framework (Nembhard and Kao, 2002). In the context of detecting *dynamical* nonstationarity, it is important to note that the notion of *weak stationarity,* found in the literature on linear time series analysis, is not considered. Weak stationarity only requires statistical quantities up to the second order (such as the *standard deviation* and *mean*) to stay constant, which in a nonlinear setting, is certainly inadequate (Kantz & Schreiber, 1997).

Unlike the detection of changes in nonlinear dynamical systems, change detection for linear systems has been studied extensively. The most complete reference is probably the work done by Basseville and Nikiforov (1993). Their work covers various change detection algorithms. This includes elementary control charts (e.g. Shewhart, geometrical moving average, finite moving average, etc.), filtered derivative algorithms, CUSUM algorithms, Bayes-type algorithms and generalized likelihood ratio (GLR) algorithms, to name a few. To apply these techniques to nonlinear systems, the data have to be linearized using nonlinear filtering methods. These linearization techniques

are not always applicable and often complicates the change detection exercise significantly (Azimi-Sadjadi & Krishnaprasad, 2002)

There are several reasons why these linear approaches are poor. Firstly, the statistics used is arbitrary and *not related to any geometrical properties of the attractor*, which is the interesting element when analyzing dynamical time series data. Such arbitrary choices are not particularly informative, unless the particular statistic estimates a parameter considered physically or dynamically important, and their power against various sorts of nonstationarity may vary greatly.

Secondly, by simply applying such statistics, the significance of differences may be greatly overestimated. This is because observed dynamical time series data are far from uncorrelated, while the simple, classical estimations of confidence rely heavily on the concept of independent observations. If one would measure the empirical means for different parts of a chaotic data set and do a classical t - test (Vining, 1998) for their equality, the null hypothesis of stationarity will quite often be falsely rejected – even when the data come from noise free stationary experiments or well-known systems such as the Lorenz attractor. Such methods do not reliably diagnose the intuitive concept of dynamical stationarity that one would typically imagine (Kennel, 1996).

Once a suitable statistic has been identified, it must be implemented in a way as to determine the "exact" time at which the process parameters changed. This will enable one to divide the data into different segments, each with similar dynamical behaviour (*signal segmentation*), resulting in better modelling and ultimately better control of the process. Rapid detection of parameter changes will also allow *online*[2] *change detection*, which is important for effective monitoring and control of chemical processes.

A number of statistical tests for nonstationarity in time series data have been proposed in the literature. Most of the tests are based on the following idea:

1) The time series under investigation is divided into a number of segments.

2) A certain parameter or statistic is calculated for each segment of the time series.

[2] Continuous monitoring of data from a process.

3) The statistic(s) from each segment are compared and if the observed variations are found to be significant, the time series is regarded as nonstationary.

In such an approach, the choice of statistic is very important. As stated earlier, linear statistics are not very useful. It is vital to use a statistic that characterizes the geometrical structure of the attractor.

In this work, emphasis is placed on the application of techniques (algorithms, statistics). Only techniques and/or statistics that have been shown to be reliable, and which is accessible (e.g. open source code) or can easily be implemented are considered. The following methods fall into this description and will be used to detect nonstationarity:

1) Evaluating the performance of process models

2) The method of nonlinear cross predictions

3) Analytical techniques with correlation dimension as test statistic

The first method uses model deterioration as an "informal" way to determine whether a time series is nonstationary. The second method is a more specialised change detection technique, whereas the correlation dimension statistic is even more precise and flexible, and will be the main focus when evaluating these techniques.

3.1.1 Model-Based Change Detection

Model-based detection of changing process parameters is probably one of the more basic methods of determining nonstationarity. In this approach a model is fitted to a segment of the time series data. The model is then used to predict other parts of the observed time series, while its performance is monitored. A good model should produce acceptable small prediction errors when applied to other process data with similar dynamic behaviour. However, if the prediction errors for the other segments of the time series differ significantly from the first segment, the data can be considered nonstationary.

When attempting to model process data, there only exists methods to optimally exploit either the *linear correlations* or *nonlinear determinism.* Hence, the decision is whether to model the process as linear stochastic or nonlinear deterministic (Kantz & Schreiber, 1997). The most prominent linear stochastic modelling techniques include: autoregressive (AR) models, moving average (MA) models and Markov models, as well as the use of nonlinear filters (Kantz & Schreiber, 1997; Gershenfeld, N.A. & Weigend, A.S., 1993). In this work, all models to be considered are of the following form,

$$\boldsymbol{Y}_t = F(\boldsymbol{X}_t) + \varepsilon_t, \tag{3.5}$$

where $\boldsymbol{X}$ is the input variables and $\boldsymbol{Y}$ is the output variables of the system., and ε is random variables which is independent of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

As stated earlier, the focus of this work is on process systems that have *deterministic dynamics*, or are at least *mixed systems* with a *dominant deterministic part*. When modelling nonlinear deterministic data, the observed time series is first embedded to form a d-dimensional reconstructed attractor. These reconstructed vectors are then used as *inputs* to the model building algorithms. Since time series data are discretely sampled over time, a deterministic model is always a map. In a time delay reconstruction it reads

$$s_{n+1} = F(s_n, s_{n-l}, s_{n-2l}, ..., s_{n-(d_e-1)l}) \tag{3.6}$$

where $F(\cdot)$ is the function that needs to be determined in order to predict a future value s_{n+1}. In the above equation d is the embedding dimension and l the time delay. Several nonlinear empirical model classes have been proposed to approximate F, including local linear model fitting, polynomial regression, radial basis function (RBF) neural networks, multiple-layer perceptron (MLP) neural networks, etc. (Kantz & Schreiber, 1997).

In this work, multiple-layer perceptron (MLP) neural networks are used for modelling purposes. It has been shown that they can successfully simulate or predict multidimensional chaotic data (De Oliveira et al., 2000). The reason for their success is

that MLP neural networks are *strong functional approximations*[3], which is consistent with the class of dynamical systems under discussion. MPL neural network modelling is relatively easy to use and is well supported by a number of large commercial software packages such as MATLAB.

Multiple-Layer Perceptron (MLP) Neural Networks:

Artificial neural networks, of which MLP neural networks is a special case, are computer algorithms which simulate, in a very simplified form, the ability of brain neurons to process information. A neural network is composed of interconnected units, organised in layers, which serve as model neurons. Each node in the network acts as a numerical processor with a specific activation (transfer) function. The role of the brain synapse[4] is modelled by a modifiable weight (w_{ij}^{k}) which is associated with each connection between neurons in adjacent layers (***Figure 3.3)***. In each neuron all the input weight signals are summed up, and either an excitatory or an inhibitory signal sent to the next layer of neurons. The decision of which signal is sent depends on whether or not the result of the sum has reached a certain threshold value, according to the specific activation function chosen. The transfer function is usually a smoothed step function:

$$\phi = \frac{1}{(1+e^{bX-c})} \tag{3.7}$$

The whole network thus becomes a problem of solving the function,

$$F(X) = \sum_{i=1}^{k} \frac{a_i}{\left(1+\exp(\boldsymbol{b}_i \boldsymbol{X} - c_i\right)}, \tag{3.8}$$

where the parameters a_i , b_i and c_i have to be determined by a fit.

Topologically, a MLP neural network (***Figure 3.3***) consists of an *input layer*, one or more *hidden layers* and an *output layer*. The number of nodes in the input layer is equal to the dimension of the input space (***X***), or in this case the dimension of the embedding.

[3] In strong functional approximations the optimal approximation error grows more slowly with dimension than for weak functional, e.g. linear and polynomial, approximations

[4] The synapse is the structure in the brain which is responsible for storing information

The number of nodes in the output layer is equal to the dimension of the output space (***Y***), i.e. the number of variables that needs to be predicted.

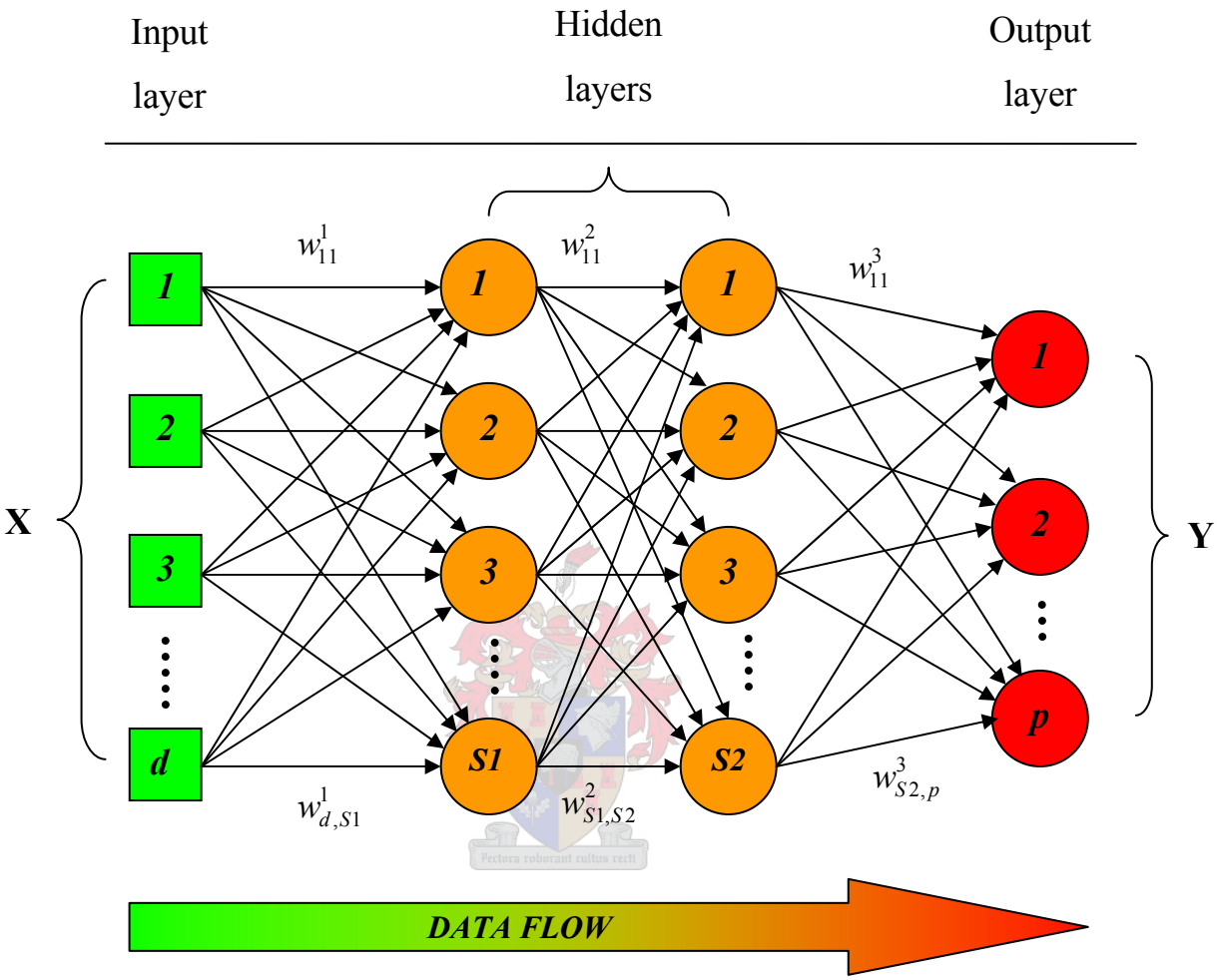


Figure 3.3: *Architectural graph of a multiple-layer perceptron neural network with two hidden layers.*

Two kinds of signals (***Figure 3.4***) are identified in a MPL neural network:

- *Functional signals* – propagate forward through the network. At each neuron through which a functional signal passes, the signal is calculated as a function of the inputs and associated weights applied to that neuron.
- *Error signals* – propagates backward through the network. Their computation by every neuron involves an error-dependent function in one form or another.

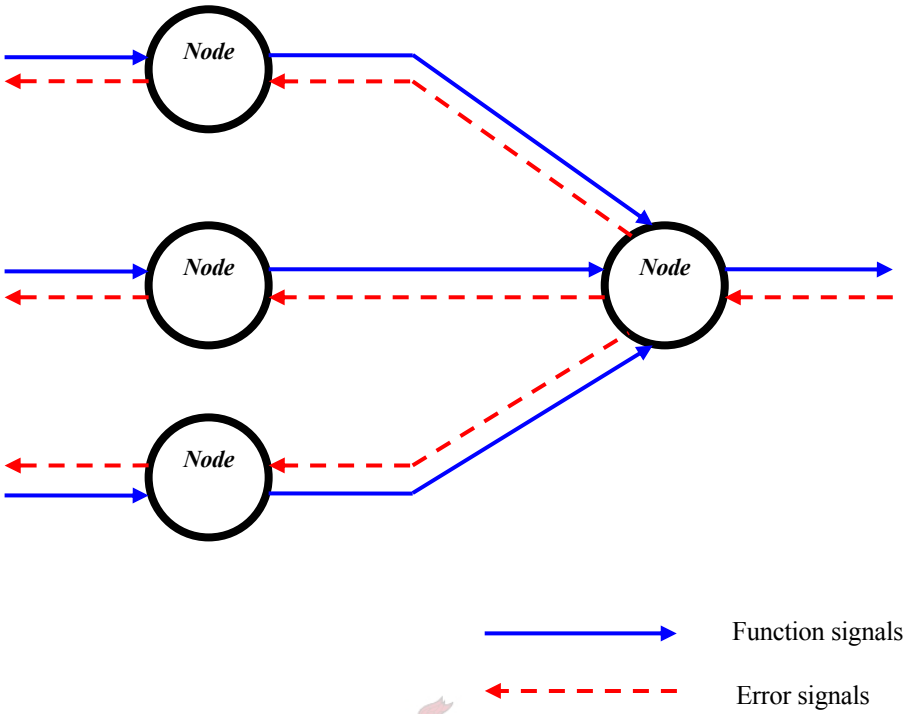


Figure 3.4: *Illustration of the directions of two basic signal flows in a multiple-layer perceptron: forward propagation of function signals and back propagation of error signals.*

The standard way to determine the model parameters, in other words the optimised weights (w_{ij}^{k}), is by training the neural network. The network is trained by feeding it with known time series data and minimising the prediction error with the *error backpropagation algorithm*. Supervised feedforward MLP neural networks learn from examples of actual data. The weights of the neuron connections are determined by presenting the network with an adequate (depending on the process system) set of actual input-output values, called the *training data set*. The output values from the network is compared with actual "unseen"[5] data from the same system by means of some *error* or *cost function*. In this work, the sum square error of the model output, defined by

[5] Data that were not part of the data set used to train the network. Comparing the network output with "seen" data would yield illusory low error results

$$E_{\theta,Z} = \frac{1}{n}\sum_{t=1}^{n}\tfrac{1}{2}r_{t,\theta}^2 , \tag{3.9}$$

is used; where θ denotes the particular model parameters and $\boldsymbol{Z} = [\boldsymbol{X}\,\boldsymbol{Y}]$ the input/output space of the particular data set. The symbol $r_{t,\theta}$ is the prediction error when the output from the neural net model, $\hat{y}_{t,\theta}$, is compared with the actual observed values from the system data, y_t. It is calculated as follow:

$$r_{t,\theta} = y_t - \hat{y}_{t,\theta} \tag{3.10}$$

The error is minimised by adjusting the weights according to the backpropagation algorithm, which corresponds to the gradient decent procedure with the inclusion of an inertial term to accelerate the convergence (see De Oliveira et al., 2000). The Levenberg-Marquart algorithm (Levenberg, 1944; Marquart, 1963) is used in this work, as it converges faster than most other training algorithms (Barnard & Aldrich, 2000).

The number of model parameters in a model structure is known as the *model order* (d_M). For a MLP network the model order depends on the dimension of the input (d) and output (p) spaces, and also the number hidden nodes (S):

$$d_M = S(d + p) \tag{3.11}$$

To determine the number of hidden nodes (S) in a nonlinear model, an approach, which starts with a subclass of model structures and then optimises the model order against Rissanen's minimum description length (MDL) calculated for each model structure, is used (Judd & Mees, 1995). This technique encodes the model parameters and model error as a bit stream of information. Under the assumption that a more complex model and a larger modelling error will need more bits to encode, the model structure with the lowest description length (MDL) will be the optimal one. The advantage of this approach is that it presents a formalised structure to determine the model order; in contrast with other methods were "overfitting"[6] usually occurs. The minimum description length (MDL) can be approximated by the simpler Schwartz Information Criterion (SIC),

[6] Implementing a model of too high an order which may fit the training data well, but generalise poor as result of the noise component in the data.

$$SIC = N\sum_{i=1}^{p}\log\left(\frac{E_{\theta,Z_i}}{N}\right) + d_M \log(N), \tag{3.12}$$

where the summation spans the set of mean square error of each component of the multi dimensional time series. Solving for ***Equations (3.11)*** and (***3.12)*** yields the model order (d_M) and the number of hidden nodes (S) in the network.

Testing for nonstationarity by validating the model:

Model validation is a crucial step in any model building procedure, as it ensures the reliable application of the model on new observations from the same data set or process. In this approach, model validation is used to test for nonstationarity. For a time series (i.e. process) to be stationary, the estimated model, trained on data from a specific segment of the recorded time series, should be able to do a faithful simulation (prediction) of observations from any other segment of the time series ***(Figure 3.5).*** In other words, the prediction error of the model should statistically be the same for all segments of the time series. The fitness of the model can be tested with the R^2-statistic,

$$R^2 = 1 - \frac{\sum(y-\hat{y})^2}{(n-1)\sigma_y^2}, \tag{3.13}$$

where y is the actual observations, $\hat{y}$ is the model simulated (predicted) values, σ is the standard deviation of y and n is the length of the predicted data set.

Another way of testing the model fitness is to plot the actual observations (y) and the predicted values ($\hat{y}$) on the same graph, and visually inspect the result. Model validation is usually based on a one-step[7] prediction, but additional insight into the model fitness can be gained by doing the more rigorous free-run[8] prediction (Small and Judd, 1998).

[7] The model is only expected to predict one time step into the future.

[8] In contrast with one-step prediction, the model is expected to predict multiple time steps into the future, which is more challenging.

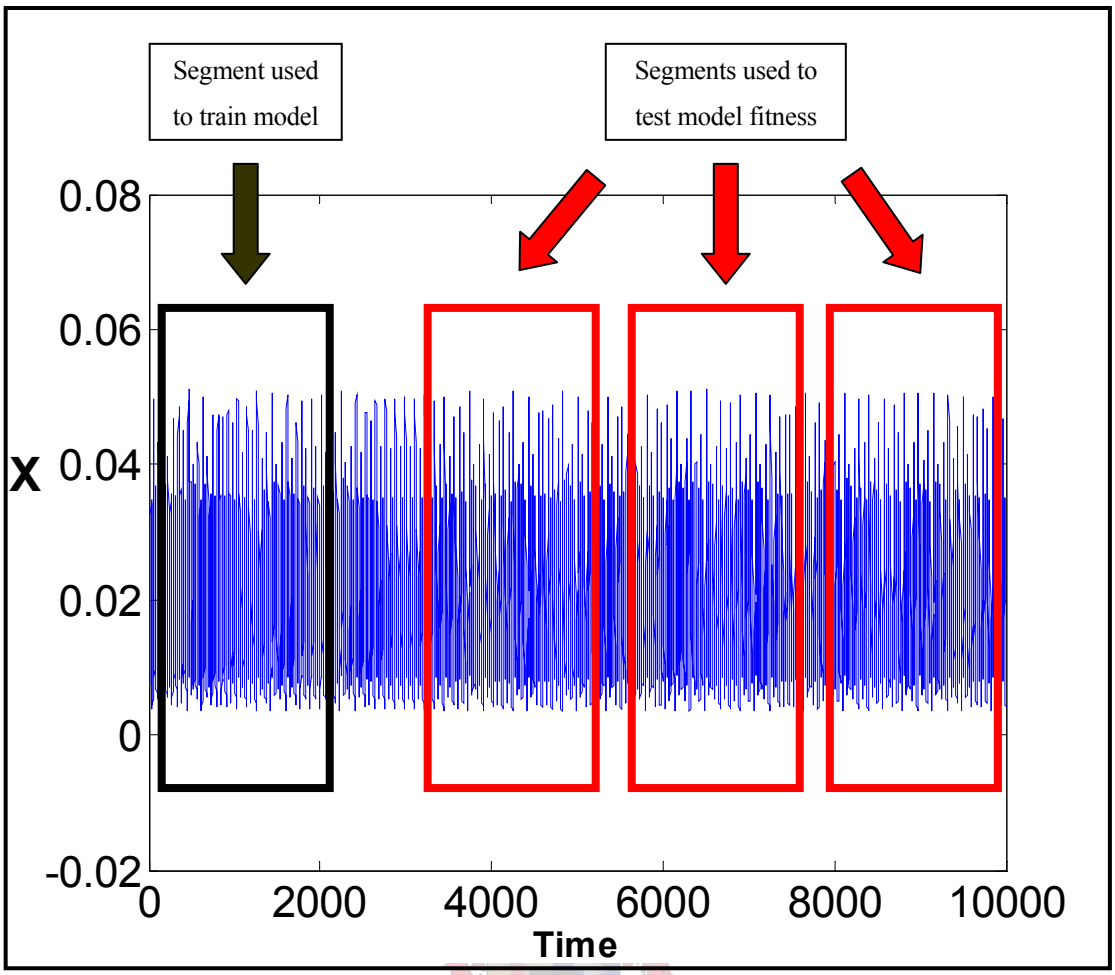


Figure 3.5: *Illustration of the model-based approach to detect nonstationarity*

3.1.2 Probing Nonstationarity Using Nonlinear Cross Prediction

A more formal test for stationarity is proposed by Schreiber (1997). The test checks for the compatibility of nonlinear approximations to the dynamics made in different segments of the time series. The segments are compared directly and not by way of statistical parameters. He claims that this approach provides detailed information about episodes with similar dynamics during the measurement period and allows for a detailed analysis of physically relevant changes in the dynamics.

The approach is based on the similarity between parts of the time series themselves, rather than the similarity of parameters derived from the time series. The similarity is

quantified by evaluating the *nonlinear cross-prediction error*[9] of different segments of the time series. Schreiber (1997) claims that the concept is particularly useful in cases were nonstationarity causes the shape of the dynamic attractor to change, while most dynamical invariants only shows slight change.

Say one has a time series $\{s_n; n = 1,...N\}$ which is split into contiguous segments of length r and with the *i*-th segment being $S_i^r = \{s_{(i-1)r+1},...,s_{ir}\}$. Traditionally, a statistic γ_i is now calculated for each segment and compared with each other, or alternatively with γ calculated from the full sequence, to see if there is any significant change in the value to suggest nonstationarity. Schreiber (1997), however, follows a different approach and use statistics defined on pairs of segments,

$$\gamma_{ij} = \gamma(S_i^r, S_j^r). \tag{3.14}$$

By using the statistic, γ_{ij}, on pairs of segments, the number of parameters calculated for a fixed number of segments increase from N/r to $(N/r)^2$. It can be argued that a lot of redundant information is gained for the purpose of statistical testing, since γ_{ij} for different segment pairs (i, j) are not expected to be independent. Schreiber (1997), however, have shown that this will allow the detection of different and more hidden kinds of nonstationarities. It will reveal a more detailed picture about the nature of the changes and, in particular, will enable one to locate segments of a nonstationary time series which are similar enough and which can be analysed together. The test exploits the information contained in the relative statistics, $\gamma(S_i, S_j)$, in addition to the information contained in the diagonal terms, $\gamma(S_i, S_i)$. The diagonal entries are expected to be systematically smaller. They represent in-sample errors since the training and test sets are the same.

In principle, the statistic $\gamma(S_i, S_j)$ can be any quantity which is sensitive to the dynamical differences in S_i and S_j respectively. It is important to use a statistic that is

[9] The nonlinear cross prediction error quantifies the predictability of one segment using another segment as database.

robust, not too sensitive to the noise level in the data and from which stable estimates can be obtained on rather short segments S_i , S_j . A statistic that meets these criteria is the *error* of the *nonlinear prediction* algorithm. Usually stable results can be obtained from a few hundred points. The main advantage of this statistic, however, is its ability to calculate cross-correlations. In this case the statistic is known as the ***nonlinear cross-prediction error***, γ_{ij} .

Let $X \equiv \{x_n, n = 1, N_X\}$ and $Y \equiv \{y_n, n = 1, N_Y\}$ be two time series, or segments of a time series. If d is specified as the embedding dimension, the embedding vectors, $\{\vec{x}_n, n = d, ..., (N_X - 1)\}$ and $\{\vec{y}_n, n = d, ..., (N_Y - 1)\}$, can be formed from the respective time series. The idea is to, for each $\vec{y}_n$, predict one step into the future, i.e. y_{n+1}, but by using X as a database. A locally constant approximation to the dynamics relating $\vec{x}_n$ and x_{n+1} yields the estimate

$$\hat{y}_{n+1}^X = \frac{1}{\left|U_\varepsilon^X(\vec{y}_n)\right|} \sum_{\vec{x}_{n'} \in U_\varepsilon^X(\vec{y}_n)} x_{n'+1} \,. \tag{3.15}$$

Here $U_\varepsilon^X(\vec{y}_n) = \{\vec{x}_{n'} : \|\vec{x}_{n'} - \vec{y}_n\| < \varepsilon\}$ is an ε-sized neighbourhood of $\vec{y}_n$, formed within the set X. The term $\left|U_\varepsilon^X(\vec{y}_n)\right|$ denotes the number of elements in that neighbourhood. For isolated points with empty neighbourhoods the sample mean of the segment X is used as an estimate for $\hat{y}_{n+1}^X$. The root mean squared prediction error, $\gamma(X,Y)$ of the sequence Y, given X, is defined by

$$\gamma(X,Y) = \sqrt{\frac{1}{N_Y - d} \sum_{n=d}^{N_Y - 1} \left(\vec{y}_{n+1}^X - y_{n+1}\right)^2} \,. \tag{3.16}$$

Test methodology:

It is expected that for a stationary time series $\gamma(S_i^r, S_j^r)$ would be independent of i and j, unless the coherent time of the process is longer than the length of the segments(r). So if there is variability in the time series on scales that is longer than r, which can be

caused by either slow parameter drift or changing parameters, the diagonal terms for the nonlinear cross-prediction error $\gamma(S_i^r, S_i^r)$ will be typically smaller than those with $i \neq j$. The results are viewed as a mutual error prediction map in the form of a 3-D surface plot, ***Figure 3.6,*** or a 2-D colour coded surface, ***Figure 3.7***.

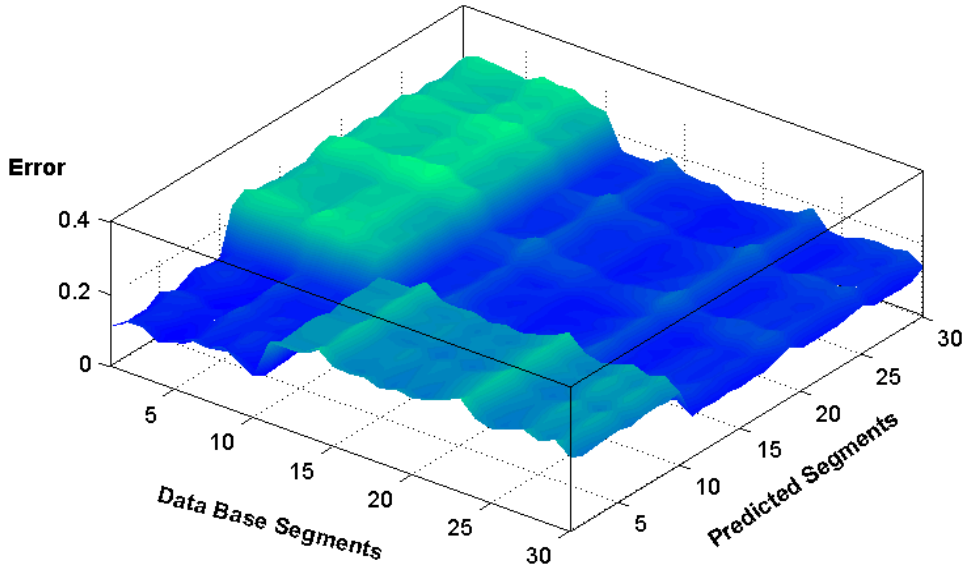


Figure 3.6: *Example of a typical 3-D surface plot for mutual cross-prediction errors.*

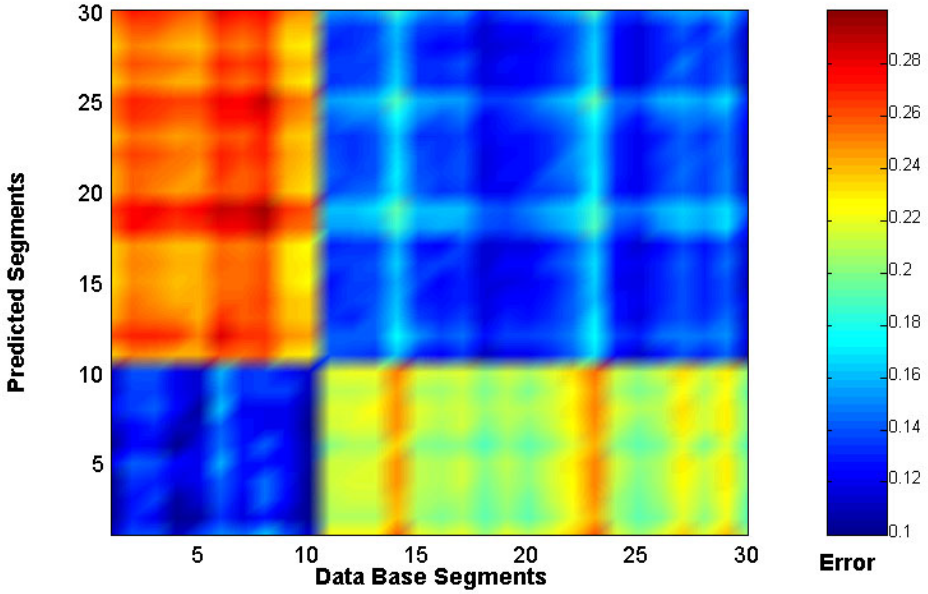


Figure 3.7: *Example of a typical 2-D colour-coded mutual prediction map*

The change in height (***Figure 3.6***), or change in colour (***Figure 3.7***), suggests that there is an explicit change in the prediction errors round about segment 10. Segments 1 to 10 are useless for predictions in segments 20 to 30, and vice versa. This indicates a change in process parameters at the time corresponding with segment 10. An element of confusion when interpreting the results is the possible asymmetry of γ_{ij}. In general $\gamma(X,Y) \neq \gamma(Y,X)$. If the attractor of Y is embedded within the attractor of X, data points in Y can be predicted with confidence using X as a database. However, Y does not contain enough information to predict all data points in X. This asymmetry, although sometimes confusing, can provide valuable insights into different kinds of nonstationarity. To simplify matters, it is possible to use a symmetrized statistic such as $\gamma_{ij} + \gamma_{ji}$, but then this advantage is lost.

3.1.3 Exploiting the Correlation Dimension as a Test for Nonstationarity

The change detection technique preferred by the author, involves the correlation dimension (d_c) as test statistic. The reason for this is because the correlation dimension (d_c) is a quantity that suffers considerably from nonstationarities in the data (Kantz & Schreiber, 1997). The theory behind correlation dimension (d_c), as well as its merit as a test statistic, were thoroughly discussed in ***Section 2.5.3***. This section only revisits some of the main issues concerning correlation dimension (d_c) calculations, and explain the approach to detect dynamic change in time series data.

Remember that for a process to be stationary there must be no significant changes in the geometrical shape of the attractor. Judd's (1992) correlation dimension algorithm produces a curve that provides information about the attractor dimension (i.e. geometrical shape) on different length scales. It therefore *characterizes the topology of the attractor,* which allows the detection of geometrical changes in the attractor, and therefore dynamical changes in the process. This is in sharp contrast with the Grassberger-Procaccia algorithm (Grassberger & Procaccia, 1983) which only gives a binary result. Many researchers (such as Hively et al., 1999) have attempted to use this binary result to detect dynamical change, without much success, and therefore

discredited correlation dimension as a useful statistic. In this work, it will be shown that the $d_c(\varepsilon_0)$-curves calculated through Judd's algorithm is especially useful for the detection of dynamical change.

Test Methodology:

There are a number of ways to use the $d_c(\varepsilon_0)$-curves to detect nonstationarities in a time series. A fairly obvious method is to calculate the $d_c(\varepsilon_0)$-curves for two halves of the time series. If the $d_c(\varepsilon_0)$-curves differ from each other, with respect to either *the region they occupy on the* $d_c(\varepsilon_0)$-ε *chart* or their *shape*, the process that generated the time series can be regarded as nonstationary (***Figure 3.8***).

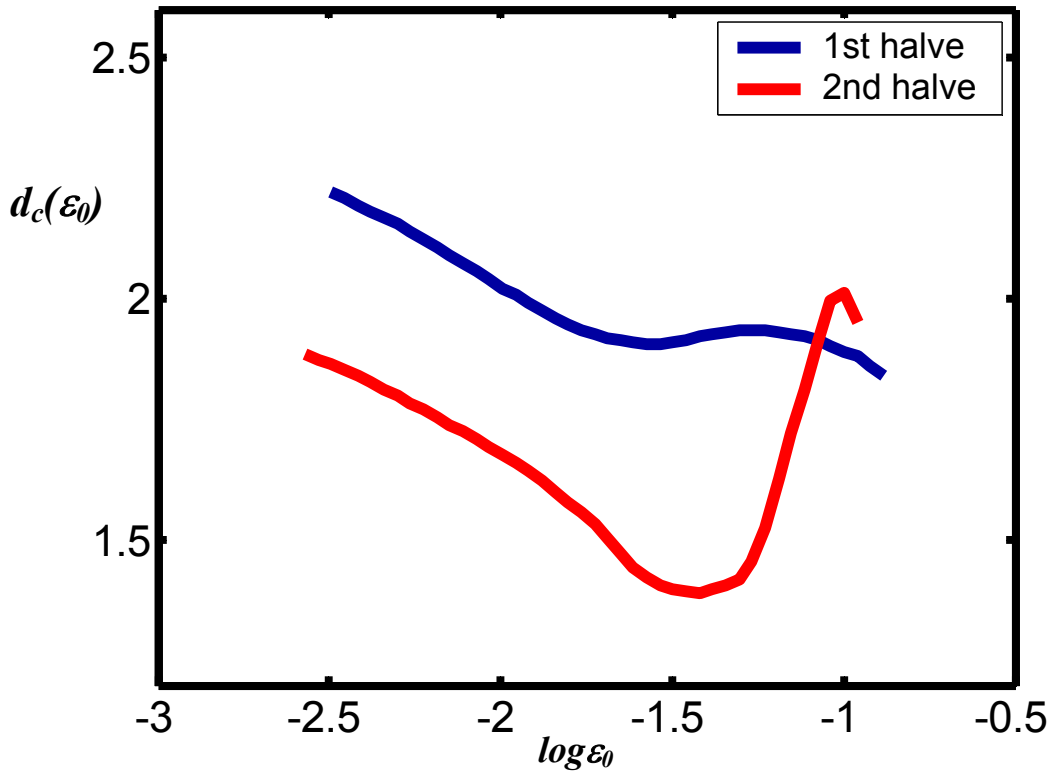


Figure 3.8: *Illustration* $d_c(\varepsilon_0)$*-curves from two halves of a time series.*

While the above approach assesses stationarity, it does not give an indication of the time at which the process parameter(s) changed, or whether there is a continuous slow parameter drift in the process. An improvement on this approach is to divide the time series into a number of segments and calculate a $d_c(\varepsilon_0)$-curve for each segment (***Figure 3.9***). An increase in the amount of segments the time series is divided into, increases the ability to identify the time at which the parameter change took place. There is a catch though. As the number of segments increase, the amount of data points in each segment decrease, and is there a growing danger that the individual segments are too short to capture the total dynamic behaviour of the process (*1st case of nonstationarity*). Furthermore, Judd's algorithm needs at least 1000 points to make a reliable estimation for $d_c(\varepsilon_0)$. So apart from keeping a segment long enough to still be considered stationary, it seems that the ultimate lower limit of a segment's size is 1000 points (This limit will be stretched somewhat in one of the case studies).

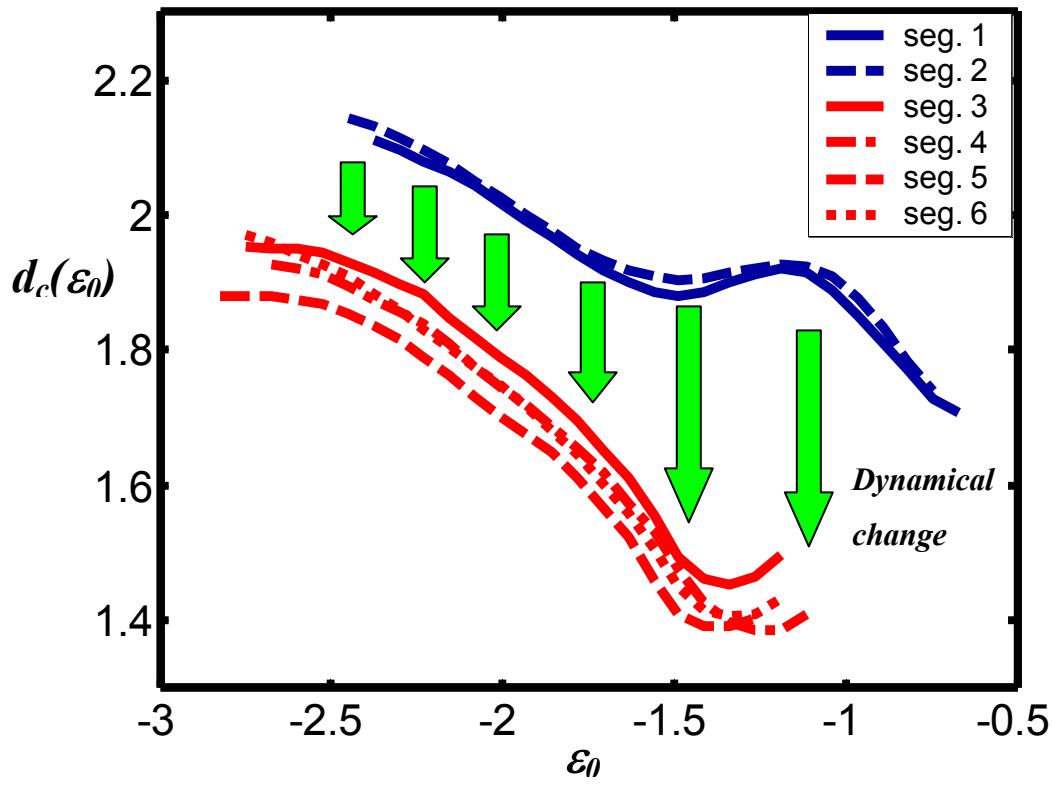


Figure 3.9: *Illustration of $d_c(\varepsilon_0)$-curves for segments of a time series. From the figure it is clear that there was a change in process parameters between segment 2 and segment 3. This is suggested by the shift of the curves, as indicated by the green arrows.*

A further improvement on the previous approach is to calculate the $d_c(\varepsilon_0)$-curves for a fixed size *moving window* (***Figure 3.10***). The *moving window approach* is useful for shorter data sets, which cannot create enough independent segments to make reliable conclusions. Even when used on longer datasets it should improve on the sensitivity[10] of the *segmentation approach*, as it allows for a larger number of realizations within the same timeframe.

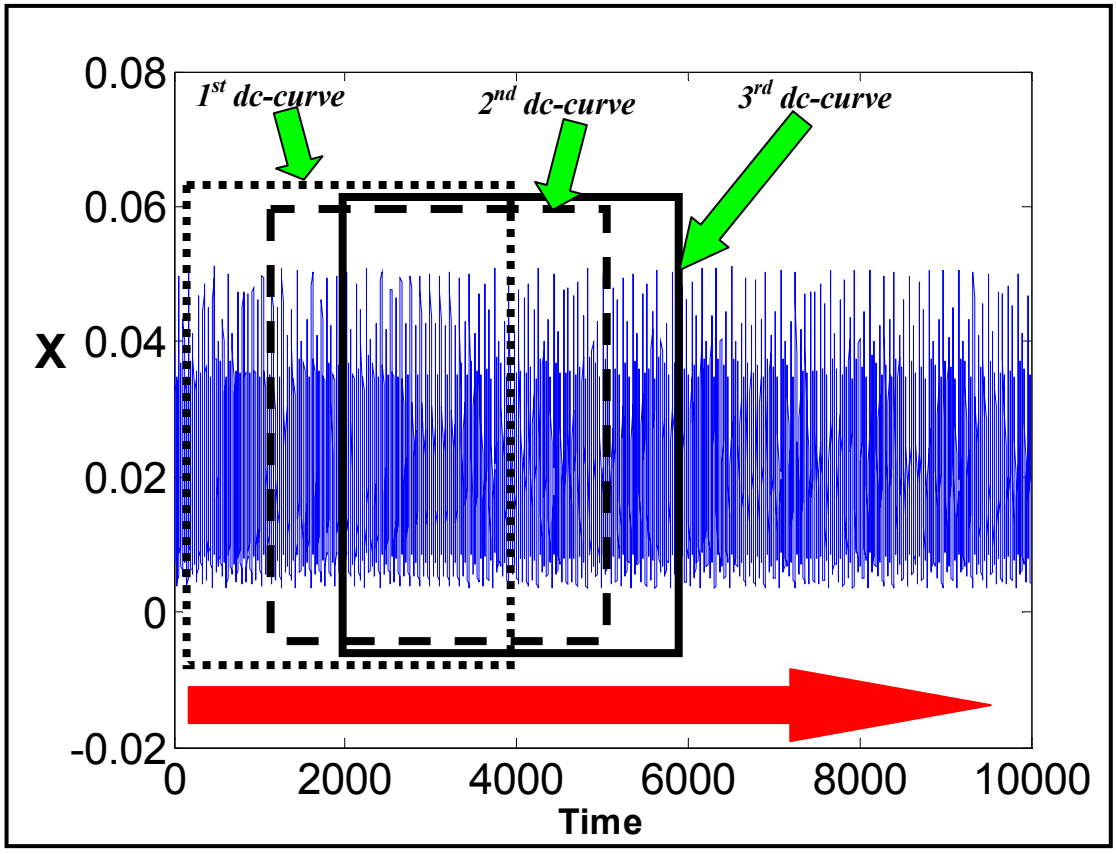


Figure 3.10: *Illustration of the moving window approach. This approach is ideal for online monitoring of changing parameters.*

The *moving window approach* can be particularly useful in the *online monitoring* of chemical processes. Here the $d_c(\varepsilon_0)$-curves are calculated continuously as new data from the process become available. Say for example that a process is being monitored where 100 new data points become available every minute, and it has been determined

[10] Sensitivity in this context means the time it takes to detect a change in parameters.

that a window size of 2000 data points will meet the requirements. It is then possible to calculate a new $d_c(\varepsilon_0)$-curve every minute, using the 100 new data points together with the last 1900 data points from the previous window. In this manner dynamic changes in chemical processes can be monitored continuously ***online***.

3.2 Change Detection Methodology

The change detection algorithms are evaluated by applying them to three case studies. The case studies were selected not only to reflect a broad range of dynamical systems, but also out of practical considerations. Each case study is approached as if the underlying dynamic behaviour of the system is unknown – which *will* be the case in most real world applications (***Figure 3.11***):

1) Time series data from the process is first classified through *surrogate data analysis*. This to acquire more information on the dynamical behaviour of the system. Knowing whether the system under investigation is *nonlinear deterministic* or *stochastic* will indicate whether more advanced change detection techniques are needed (as discussed in this work), or if traditional SPC methods will be sufficient.

2) If the system prove to be *nonlinear deterministic,* the optimal embedding parameters (time delay and dimension) for the data set are determined and the state space reconstruction done.

3) The change detection techniques are applied to the data to determine whether (and *when*) dynamic changes occurred in the process.

The change detection techniques will be evaluated on their ability to detect intrinsic changes in the dynamics of the processes, as well as the time necessary to detect these changes.

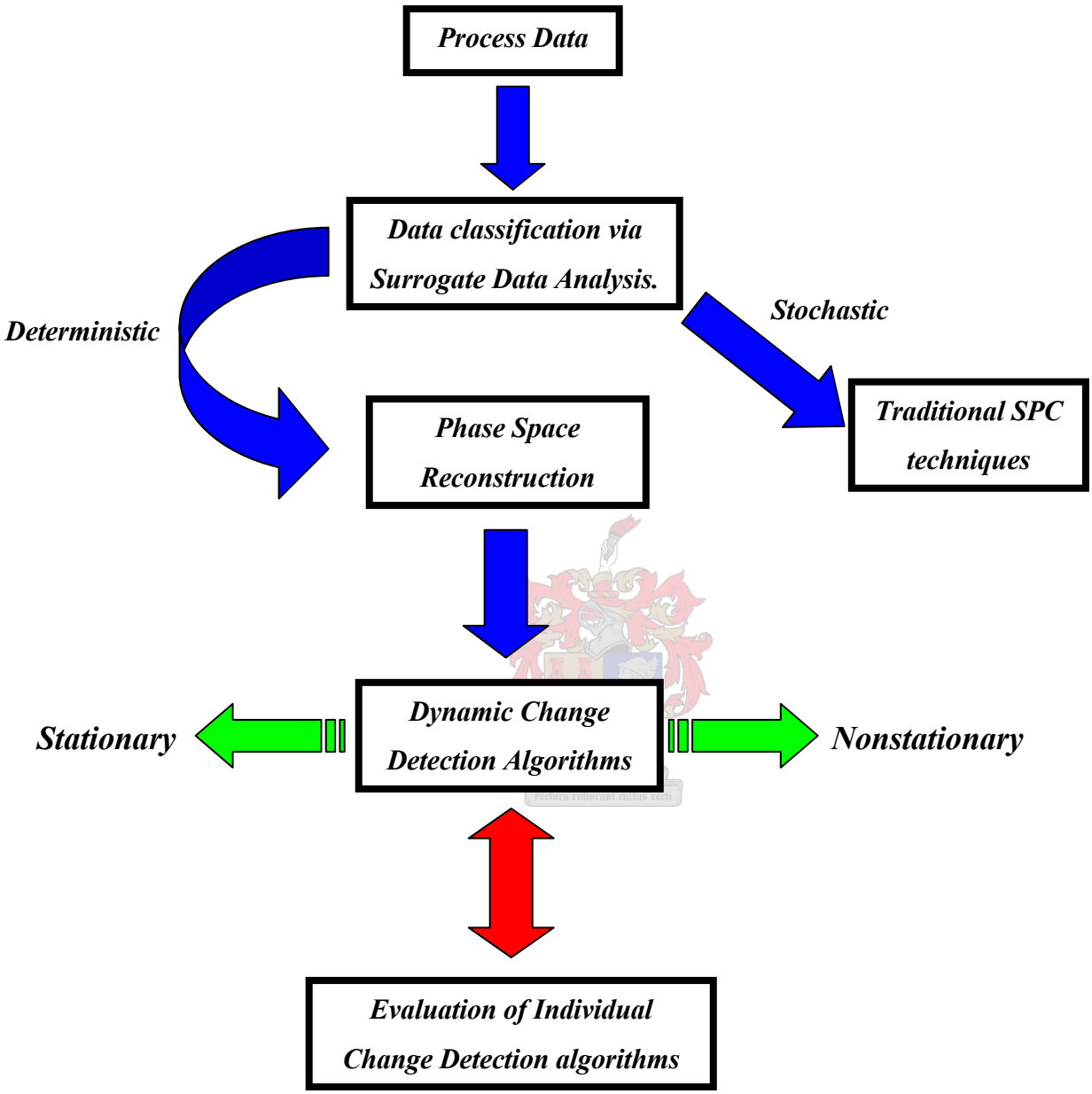


Figure 3.11: *A flow diagram of the approach to detect parameter change.*

4 CASE STUDIES

Evaluating change detection techniques using actual process data is not ideal. The major problem is that one usually do not know beforehand whether the time series data are stationary or not (This is the reason for developing nonstationarity tests!). Reliable conclusions about the performance of these techniques cannot be made if the behaviour of the data is unknown. It is for this reason that two of the case studies are based on simulated data with manually induced parameter changes. Nevertheless, because the whole point of this research is to improve the control of real-life chemical processes, it is important to include a case study based on observations from an actual process. The third case study is based on actual data from a metal leaching plant.

4.1 Autocatalytic Reactor

The first case study is a classic chemical engineering problem. The system is consists of two parallel, isothermal autocatalytic reactions taking place in a continuous stirred tank reactor (CSTR) (Lee & Chang, 1996).

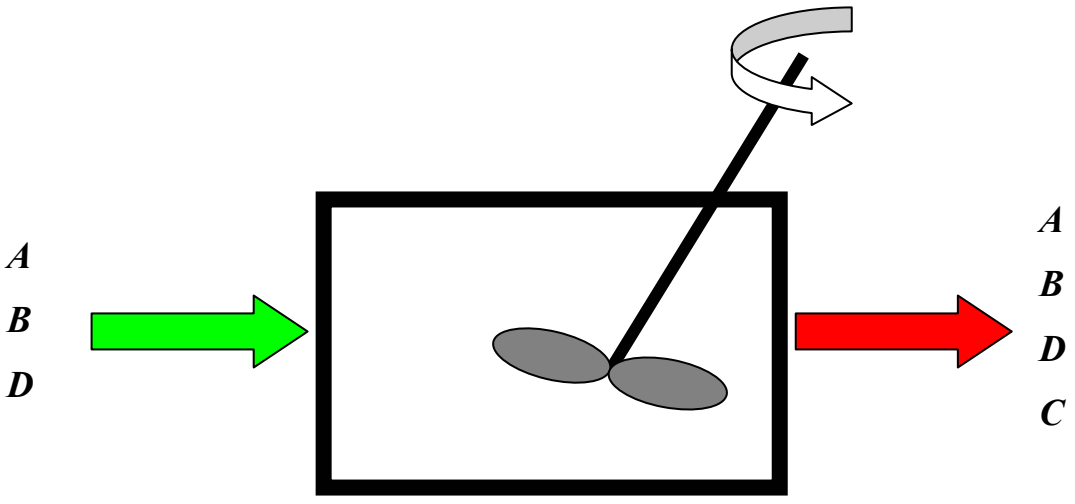


Figure 4.1: *Schematic illustration of the autocatalytic reactor.*

The kinetics of the system proceeds according to the following steps:

$$A + 2B \rightarrow 3B \tag{4.1}$$

$$B \rightarrow C \tag{4.2}$$

$$D + 2B \rightarrow 3B \tag{4.3}$$

The kinetics is governed by the following rate equations:

$$-\gamma_A = k_1 C_A C_B^2 \tag{4.4}$$

$$-\gamma_C = k_2 C_B \tag{4.5}$$

$$-\gamma_D = k_3 C_D C_B^2 \tag{4.6}$$

When the reaction is carried out in a CSTR, the system can be described by three ordinary differential equations:

$$\frac{dx_1}{d\tau} = 1 - x_1 - D_1 \cdot x_1 \cdot x_3^2 \tag{4.7}$$

$$\frac{dx_2}{d\tau} = 1 - x_2 - D_2 \cdot x_2 \cdot x_3^2 \tag{4.8}$$

$$\frac{dx_3}{d\tau} = 1 - (1 + D_3) \cdot x_3 + \gamma_1 \cdot D_1 \cdot x_1 \cdot x_3^2 + \gamma_2 \cdot D_2 \cdot x_2 \cdot x_3^2 \tag{4.9}$$

The parameters in differential equations are explained by the following relationships:

$$x_1 = \frac{C_A}{C_{A0}}, \qquad x_2 = \frac{C_D}{C_{D0}}, \qquad x_3 = \frac{C_B}{C_{B0}}, \tag{4.10}$$

$$D_1 = \frac{k_1 C_{B0}^2 V}{Q}, \qquad D_2 = \frac{k_3 C_{B0}^2 V}{Q}, \qquad D_3 = \frac{k_2 V}{Q}, \tag{4.11}$$

$$\gamma_1 = \frac{C_{A0}}{C_{B0}}, \qquad \gamma_1 = \frac{C_{D0}}{C_{B0}}, \tag{4.12}$$

$$\tau = \frac{t \cdot Q}{V}. \tag{4.13}$$

where x_i is the dimensionless concentration of C_A, C_D and C_B; D_i the Damkohler numbers for the species A, D and B; γ_i the ratios of the species in the feed; and τ the dimensionless time.

With the parameters $D_1 = 18000$, $D_2 = 400$, $D_3 = 80$, $\gamma_1 = 1.5$ and $\gamma_2 = 4.2$, the continuous autonomous system represented by ***Equations (4.9) – (4.11)*** exhibits chaotic behaviour (Lee & Chang, 1996). The system is solved by generating 10000 observations using the ODE45 subroutine in MATLAB. The time is varied from 0 to 50 using a sampling rate of 0.005. At $\tau = 50$, a parameter change is induced whereby the parameters γ_1, γ_2 slowly change (at a constant rate) from $\gamma_1 = 1.5$, $\gamma_2 = 4.2$ to $\gamma_1 = 1.55$, $\gamma_2 = 4.25$ over the next 10000 observations. At the point where $\tau = 100$, the two parameters are kept fixed at $\gamma_1 = 1.55$ and $\gamma_2 = 4.25$ for the next 10000 observations, until $\tau = 150$. The result is a time series consisting of three regions:

1) Observations $1 \rightarrow 10000$ – A stationary part were all the system parameters are kept constant.

$$D_1 = 18000$$
$$D_2 = 400$$
$$D_3 = 80$$
$$\gamma_1 = 1.5$$
$$\gamma_2 = 4.2$$

2) Observations $10001 \rightarrow 20000$ – A nonstationary part with a slow, but constant, parameter drift.

$$\begin{aligned} D_1 &= 18000 \\ D_2 &= 400 \\ D_3 &= 80 \\ \gamma_1 &= 1.5 \\ \gamma_2 &= 4.2 \end{aligned} \quad \xrightarrow{\textit{Slow Parameter change}} \quad \begin{aligned} D_1 &= 18000 \\ D_2 &= 400 \\ D_3 &= 80 \\ \gamma_1 &= 1.55 \\ \gamma_2 &= 4.25 \end{aligned}$$

3) Observations $20001 \rightarrow 30000$ – Parameters stabilize at their new values to form a second stationary part.

$$D_1 = 18000$$
$$D_2 = 400$$
$$D_3 = 80$$
$$\gamma_1 = 1.55$$
$$\gamma_2 = 4.25$$

What in fact happened to the process, from a real-life perspective, is that the feed concentrations C_{A0} and C_{D0}, after being constant for a period of time, gradually increased over time and stabilized at a new higher value; thus creating a nonstationary time series. All the calculations in this case study are based on the x_1 observations (shown in ***Figure 4.2***), as it is assumed that only the x_1 - variable could be measured.

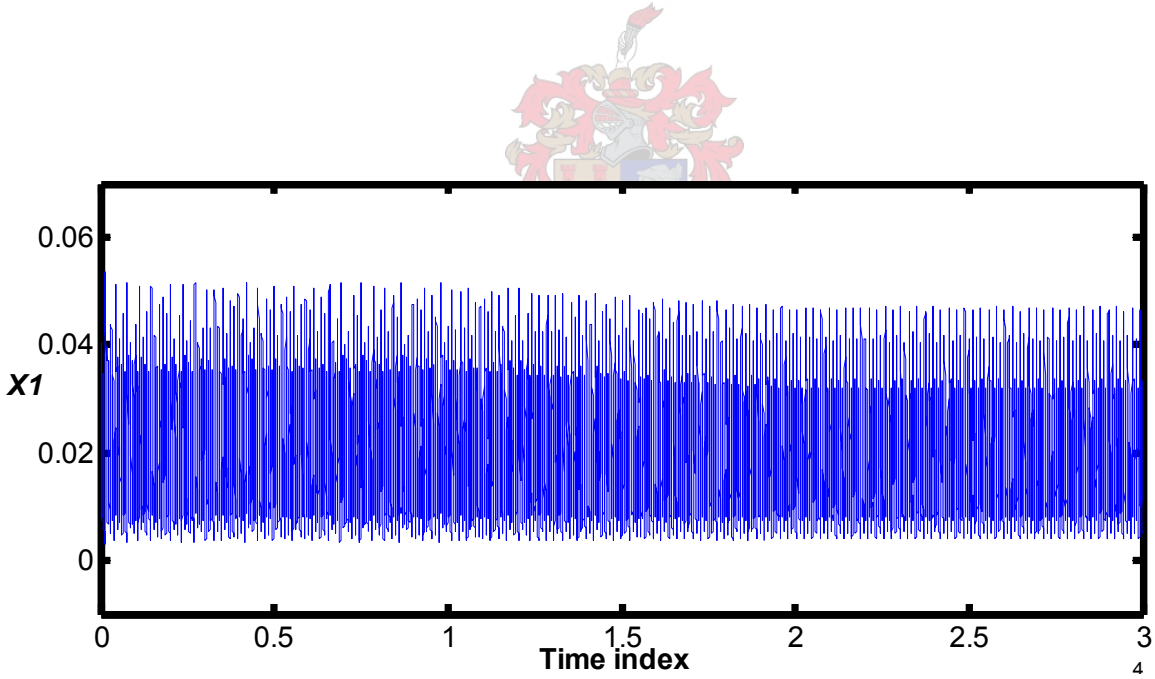


Figure 4.2(a): *All 30 000 data points from the nonstationary time series generated by the autocatalytic reaction.*

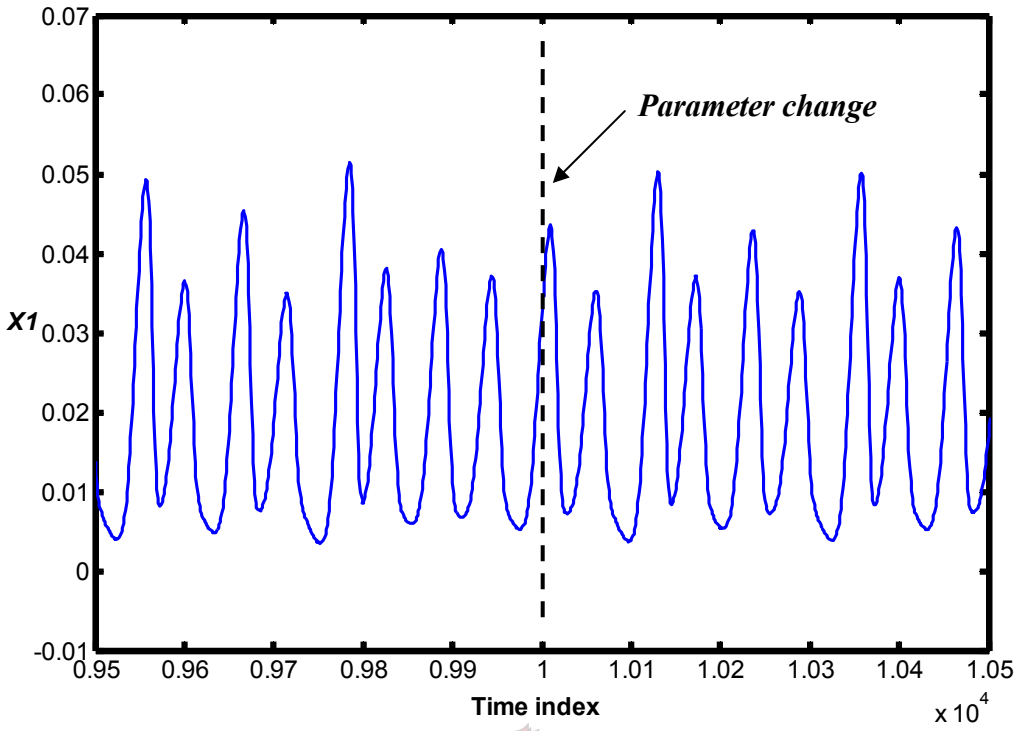


Figure 4.2(b): *Data points 9500 to 10500 from the nonstationary time series generated by the autocatalytic reaction.*

It is impossible to detect the hidden nonstationarities in time series (shown in ***Figure 4.2(b)***) visually. From ***Figure 4.2(a)*** it can be seen that the mean and standard deviation of the data remain virtually unchanged during the parameter changes. Traditional SPC methods will be unable to detect these changes. More advanced techniques, which can characterise the dynamic behaviour of the system, is needed.

4.1.1 State Space Reconstruction of the Autocatalytic System

The first step in state space reconstruction is to calculate the embedding parameters, i.e. the time delay (or time lag) and the embedding dimension. Either the autocorrelation function, or the average mutual information (AMI) criterion can be used to determine the time delay. The average mutual information statistic gives the following result (***Figure 4.3***):

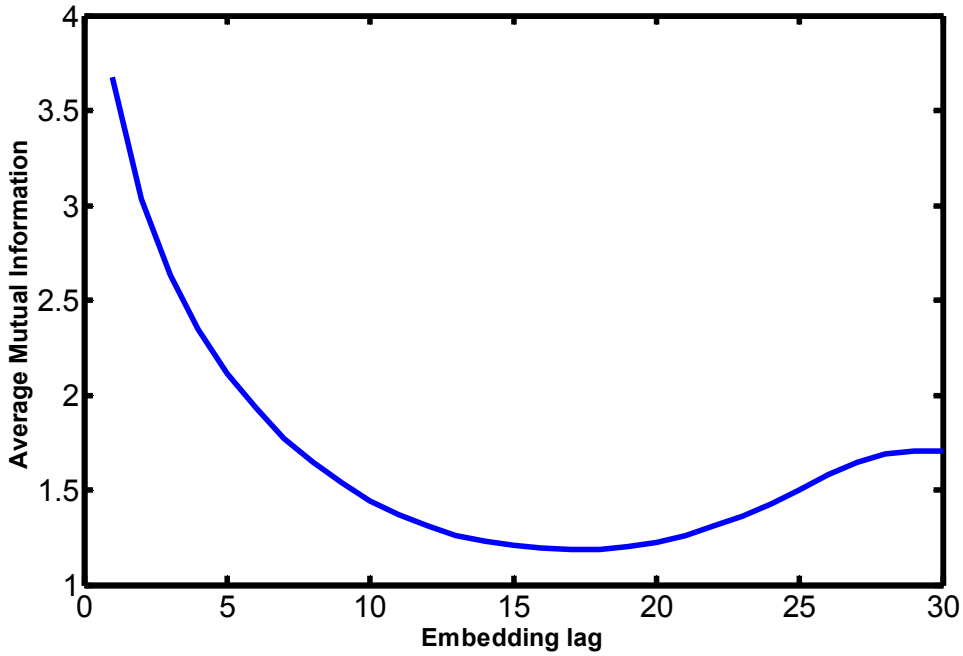


Figure 4.3: *The average mutual information, as function of the time delay, for the autocatalytic reaction.*

The most suitable time delay is where the first minimum value of the average mutual information statistic occurs (Fraser & Swinney, 1986) – in this case 17 time steps. Calculating the time delay with the autocorrelation function gives ***Figure 4.4*** as result. Closer inspection of the graph suggests 15 time steps as the embedding delay – the first value where the autocorrelation function is zero. The values for both the average mutual information and autocorrelation function are very close. This is not always the case. In principle, the AMI statistic gives better results than the autocorrelation function. The reason being that the autocorrelation function is a linear statistic, while the AMI statistic takes the aspect of chaotic behaviour into consideration (as discussed in ***Chapter 2***). The autocorrelation function is also known to sometimes overestimate the time delay, in which case it is better to use the mutual information statistic (Barnard & Aldrich, 2000).

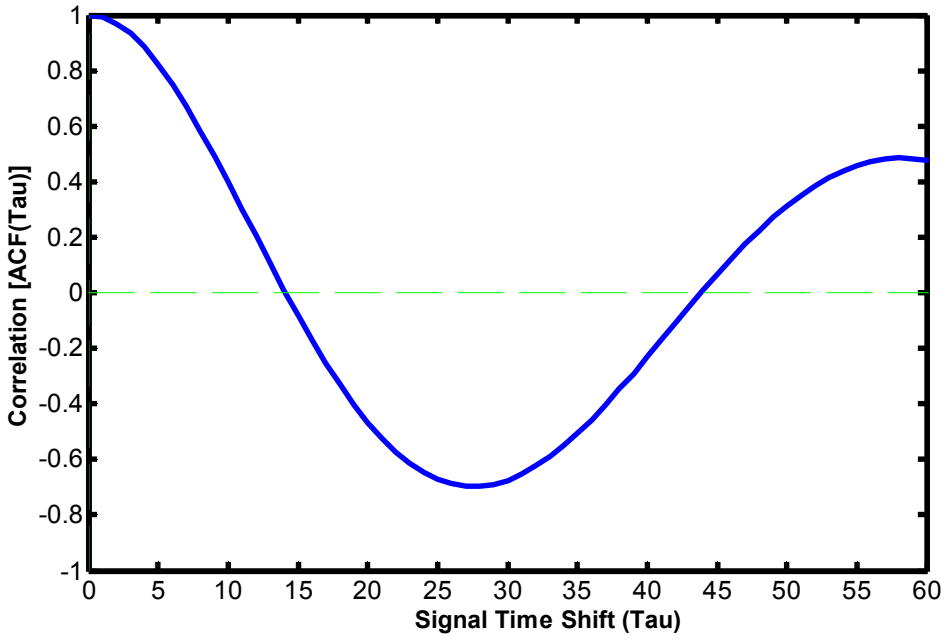


Figure 4.4: *Autocorrelation function as statistic to determine time delay for the autocatalytic reaction.*

The embedding dimension is determined by the FNN (False Nearest Neighbours) algorithm, using 17 time steps as the embedding delay. ***Figure 4.5*** shows that the fraction of false nearest neighbours is almost zero at an embedding dimension of 3.

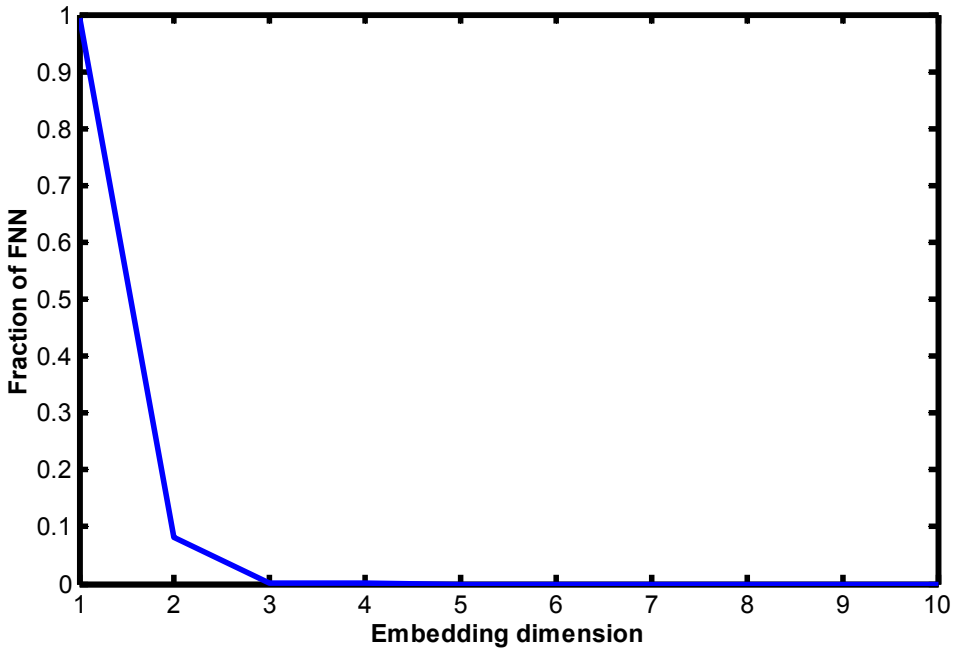


Figure 4.5: *Fraction of FNN as function of embedding dimension for the autocatalytic reaction.*

The system's attractor can now be reconstructed using a time delay of 17 and an embedding dimension of 3. The reconstructed attractor is shown in ***Figure 4.6.***

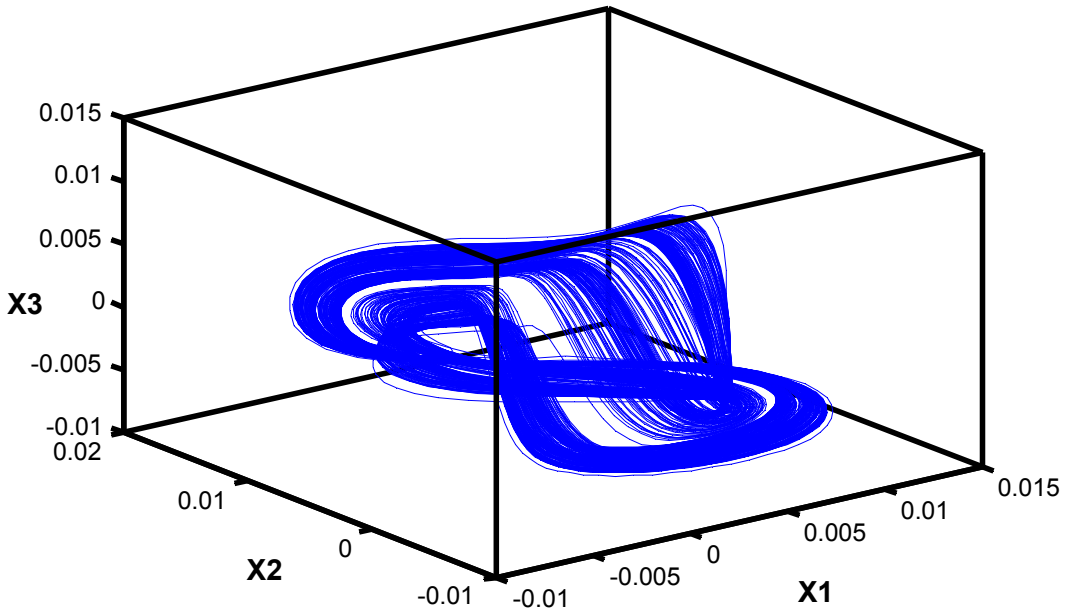
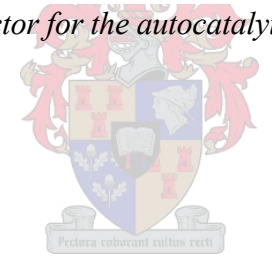


Figure 4.6: *Reconstructed attractor for the autocatalytic reaction system*



4.1.2 Surrogate Data Analysis for the Autocatalytic System

When investigating any system, it is important to know whether it exhibits deterministic behaviour, or linear stochastic behaviour. This classification will provide valuable insight into the dynamics of the autocatalytic process, and will confirm whether more advanced change detection techniques need to be used. Fifteen (15) *type 2* surrogate data sets are generated using the *Amplitude Adjusted Fourier Transform* (AAFT) algorithm. These surrogates are compared with the actual data using Judd's correlation dimension (d_c) estimate as statistic. The data sets (both the actual data and its surrogates) are embedded into 10 dimensions to ensure that the attractor from the actual data set has completely unfolded.

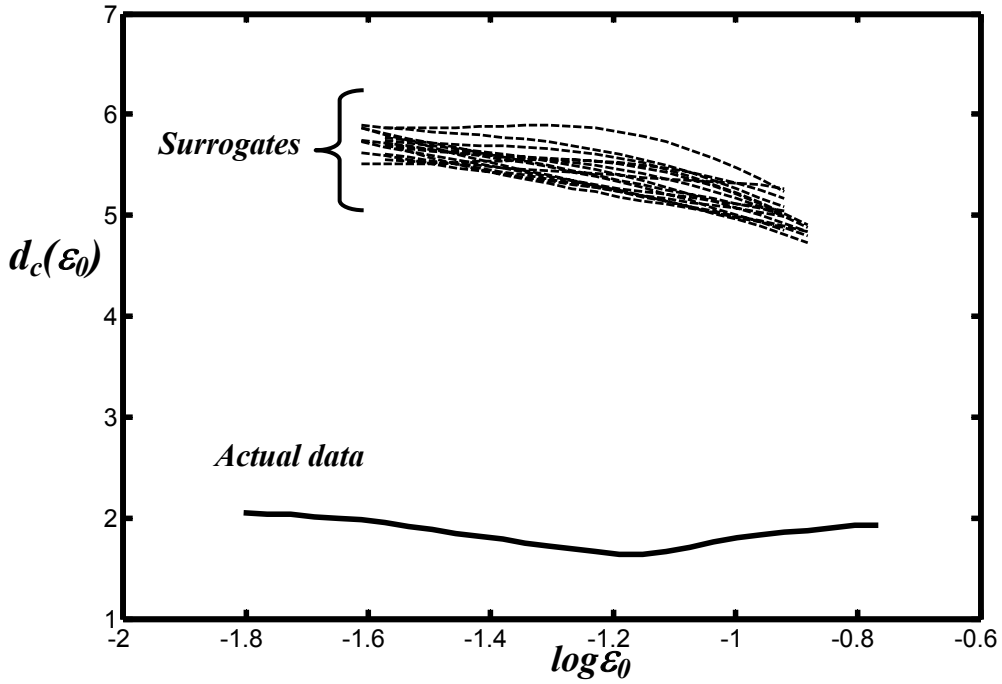


Figure 4.7: *Correlation dimension curves for surrogates and actual data for autocatalytic reaction.*

There is a definite separation between the surrogate data sets and the actual data (***Figure 4.7***). This, as well as the low dimensionality of the d_c - curve from the actual data, suggests that the system is nonlinear and governed by highly deterministic rules. The outcome is as expected, since the data were simulated from three nonlinear differential equations governing the dynamics of the system – resulting in a low (three) dimensional attractor.

4.1.3 Modelling the Autocatalytic System

Knowing that the system under investigation is most likely nonlinear deterministic, the use of MLP-neural networks will probably serve best in modelling the data. The idea is to see how nonstationarities in the data influence the possibility of modelling the

system. A deterioration in the performance of the model will serve as a test for detecting nonstationarities in the data.

To start with, a model is fitted to the first 5000 data points of the time series. The Levenberg-Marquardt algorithm is applied to estimate the model parameters. The algorithm is instructed to search for an optimum model that has between 2 and 20 hidden nodes. It is important not to "overfit" the model, as it may lead to the problems described in ***Section 3.1.1.*** The algorithm attempts to minimise the Schwartz Information Criterion (SIC) for the model. As soon as there are 3 consecutive model estimates, for which the *SIC* does not improve on the current global minimum, the parameter estimation ends. The number of hidden nodes that corresponds to this minimum SIC, is the amount used in the MPL network. ***Figure 4.8*** illustrates how the number of hidden nodes was optimised. In this particular instance, 17 nodes is the optimum.

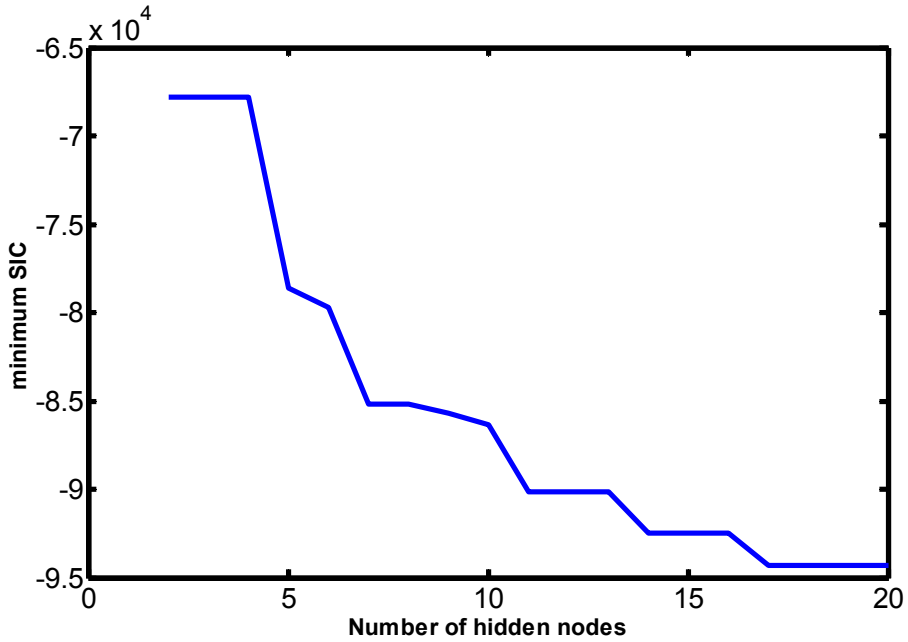


Figure 4.8: *History of the moving global minimum of the Schwartz Information Criterion versus the number of hidden nodes for the autocatalytic system.*

The model, estimated from the first 5000 data points of the time series, is now used to predict "unseen" data from the rest of the time series, while its performance is evaluated. The free-run prediction is used to validate the model, as it is a more rigorous test of the validity of the model than the one-step prediction. Since the simulated data are noise free, the one-step validation will give very low prediction errors ($R^2 \approx 1$) in

particular, and more insight will be gained by analysing the free-run predictions. This was verified by fitting the model to any part of the time series, even after the parameters had changed, and it still gave a R^2 value of essentially one.

The stationarity of the process is assessed by performing a series of free-run predictions on the different dynamic regions of the time series:

1) Data points 5000-5200 – part of the first dynamical stationary segment of the data.

2) Data points 15000-15200 – part of the segment where slow parameter drift occured.

3) Data points 25000-25200 – part of the segment where the process parameters had stabilized again.

Although the model still performs reasonably well in all three instances, it is evident (by examining ***Figures 4.9 to 4.14***), that the performance of the model deteriorates when trying to predict data outside its dynamic range. The model gives an excellent prediction of "unseen" data within the same dynamic range (***Figure 4.9***). It predicts the 200 data points without a problem, and this is verified by the low residuals[1] (***Figure 4.10***). Predictions outside the model's dynamic range become increasingly more difficult. The model cannot predict the data with the slow parameter drift as well as before (***Figure 4.11***) and the residuals are almost 10 times larger than on data with the same dynamic behaviour (***Figure 4.12***). The free-run prediction almost breaks down when predicting data from the segment with the new process parameters. It is only capable of predicting about 30 data points into the future (***Figure 4.13***) and the residuals are again almost 10 times larger than in the previous segment (***Figure 4.14***).

This deteriorating performance of the initial model indicates that there might have been some changes in the process parameters, had it been unknown beforehand. Although this is not a refined method for detecting nonstationarity, it can serve as an indication of changing parameters.

[1] The residual is the difference between the actual and predicted value.

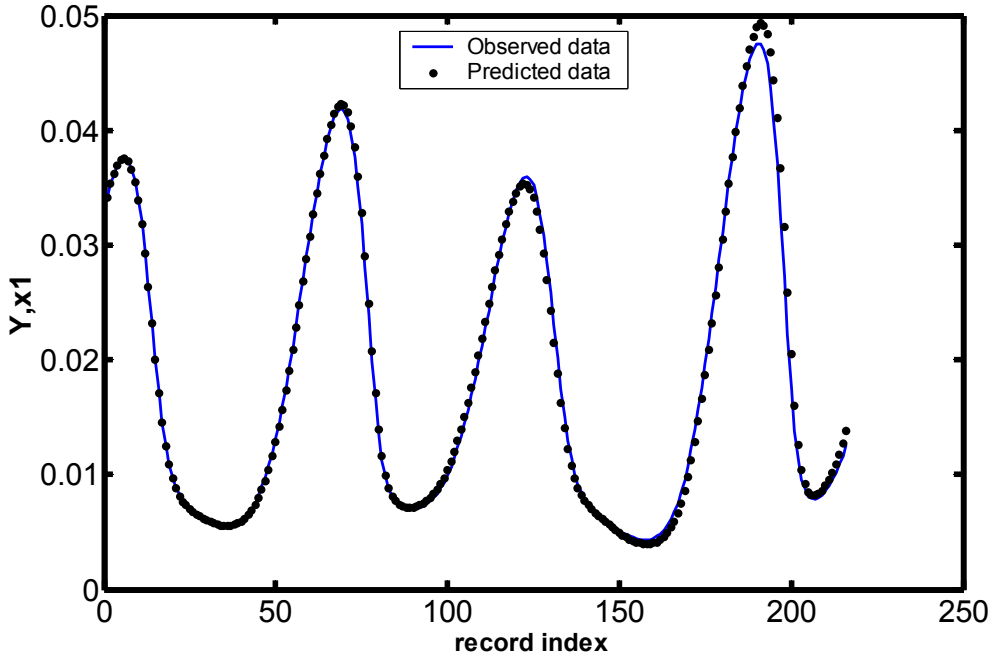


Figure 4.9: *Free-run prediction of data points 5000-5200.*

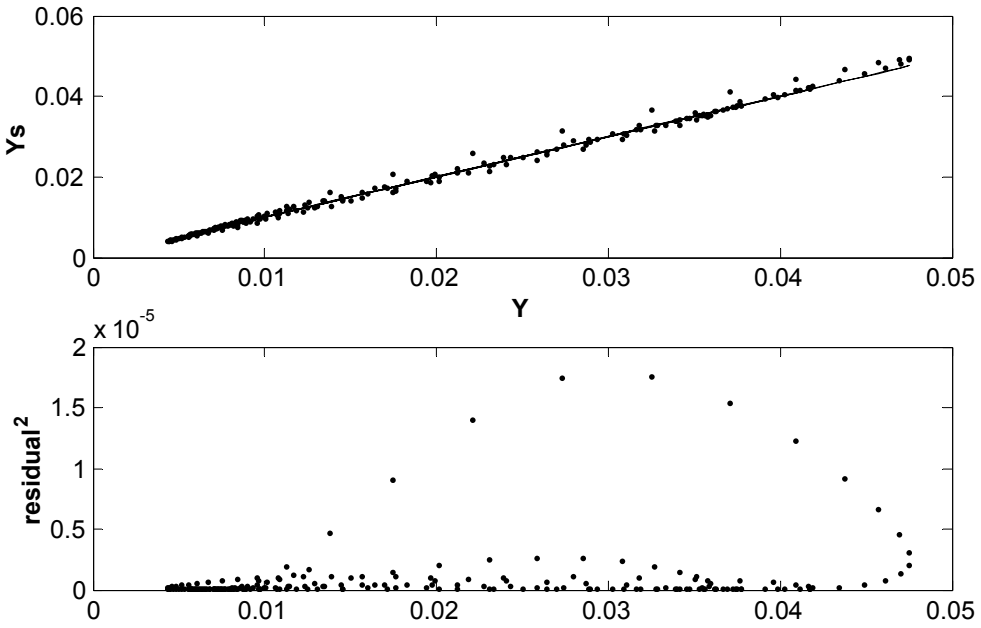


Figure 4.10: *Plot of predicted values versus actual values (5000-5200), and the residuals.*

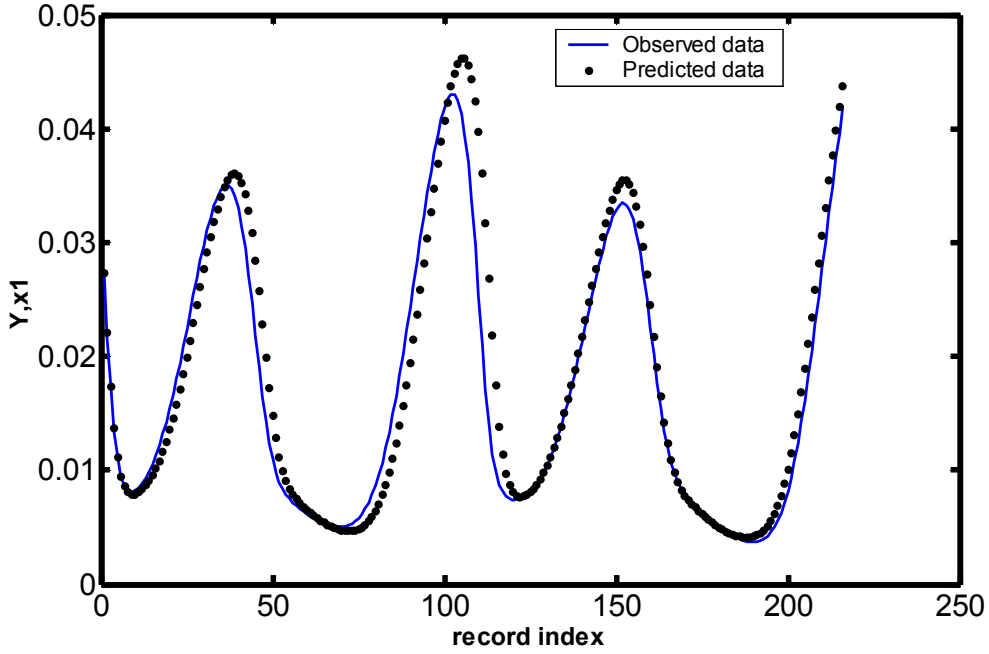


Figure 4.11: *Free-run prediction of data points 15000-15200.*

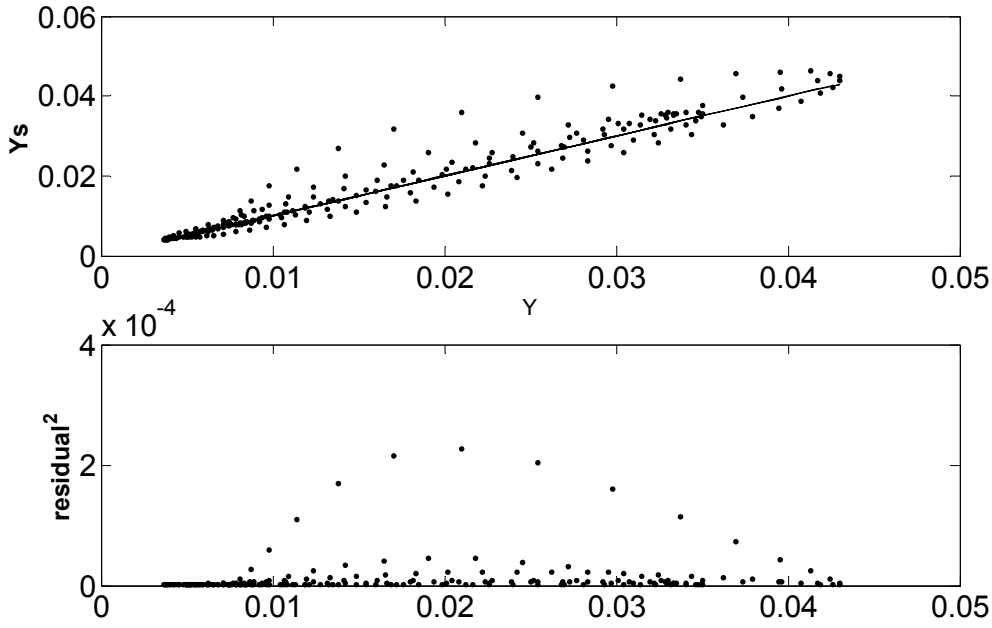


Figure 4.12: *Plot of predicted values versus actual values (15000-15200), and the residuals.*

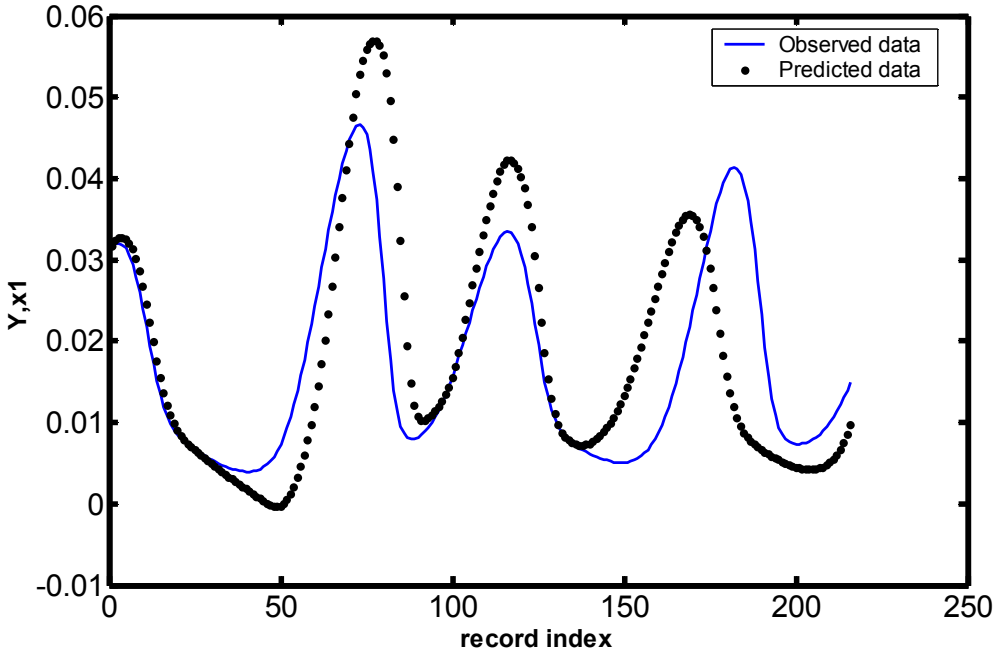


Figure 4.13: *Free-run prediction of data points 25000-25200.*

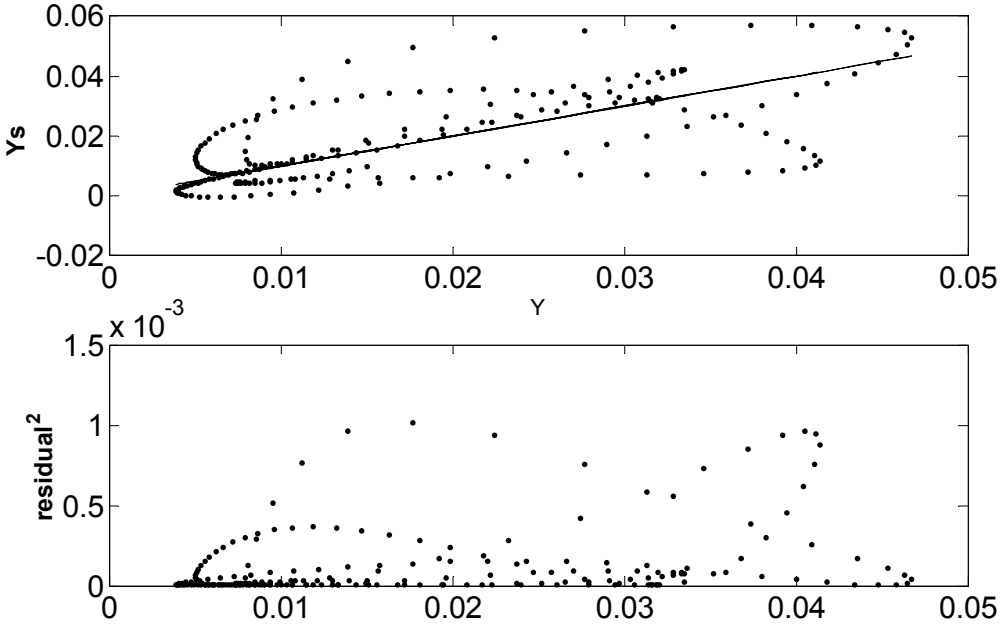


Figure 4.14: *Plot of predicted values versus actual values (25000-25200), and the residuals.*

4.1.4 Detecting Dynamic Change in the Autocatalytic Reactor using Nonlinear Cross Prediction – Schreiber's Approach

In Schreiber's (1997) approach the *nonlinear cross-prediction error* between different segments of the time series is calculated and viewed in the form of a mutual map. The time series is divided into 30 segments, each containing 1000 data points. The mutual cross-prediction errors between these segments are calculated using the previously determined embedding parameters of $l = 17$ and $d = 3$. The results can be viewed in the form of a 3-D surface plot (***Figure 4.15***) and a 2-D colour-coded surface (***Figure 4.16***). The plots of the mutual prediction errors indicate that there is a change in process parameters round about segment 15-17, which corresponds to data points 15000-17000. Segments 17-30 are useless for predicting segments 1-17, which suggests that the process dynamics from these two parts are not the same.

Something interesting is revealed by a closer inspection of the mutual prediction maps. While segments 17-30 cannot predict segments 1-17 with great success, segments 1-17 are still very useful for predicting segments 17-30. This may seem contradictory, as one usually expects that a change in parameters would result in an equally bad result both ways. In the previous chapter it was stated that, while the prediction error is mostly symmetric, in general $\gamma(X,Y) \neq \gamma(Y,X)$. The asymmetric result point towards the possibility that the attractor for the data points 17000-30000 (more or less the time where the process parameters stabilized at their new values) is somehow embedded within the attractor for the data points 1-15000 (more or less the time of the original fixed process parameters). This is confirmed by plotting the two parts of the reconstructed attractor on the same axis (***Figure 4.17***).

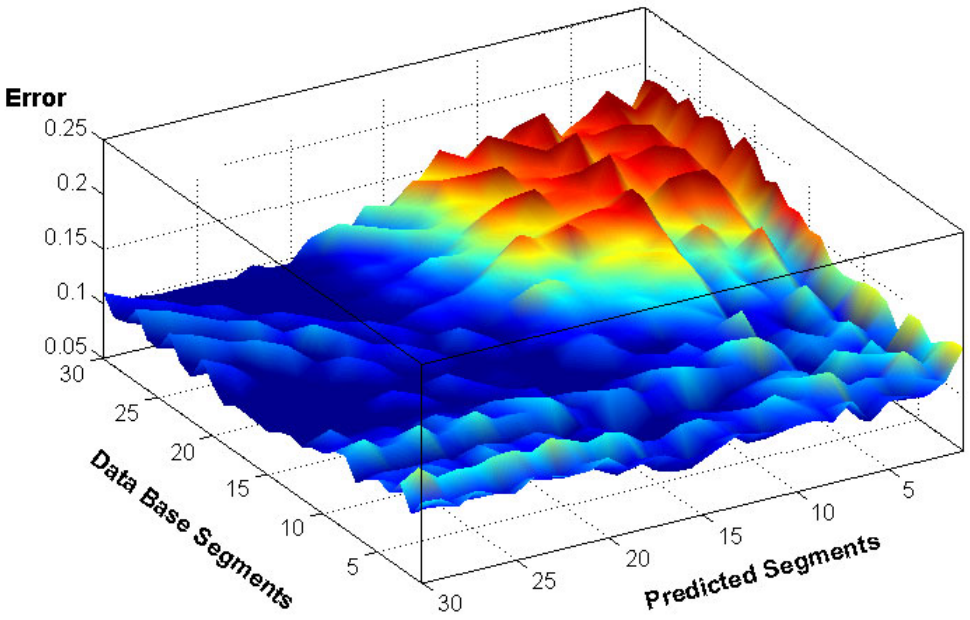


Figure 4.15: *3-D surface plot of mutual cross-prediction errors for the autocatalytic reaction.*

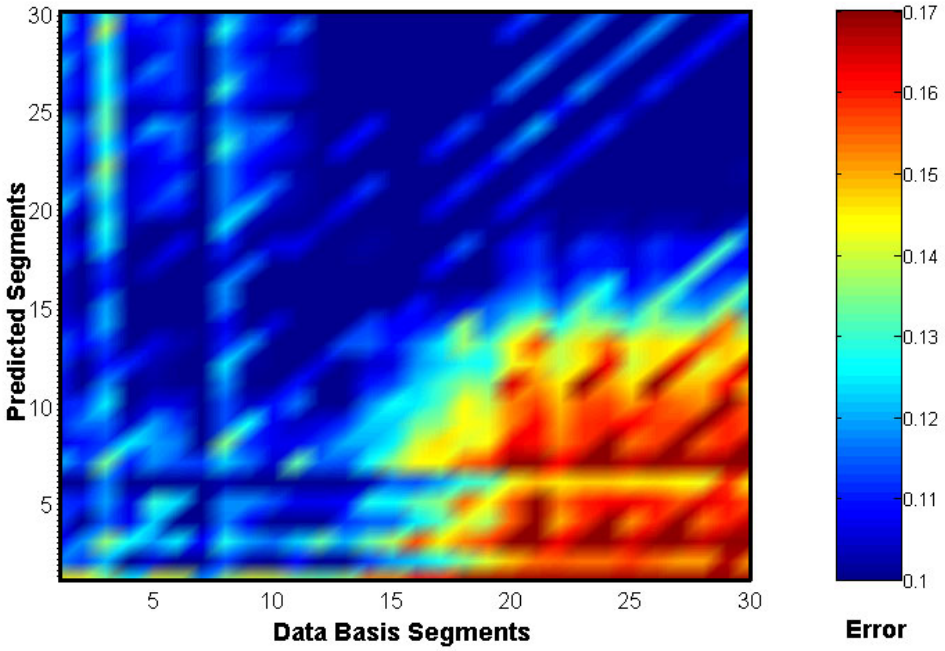


Figure 4.16: *2-D colour-coded mutual prediction map for autocatalytic reaction.*

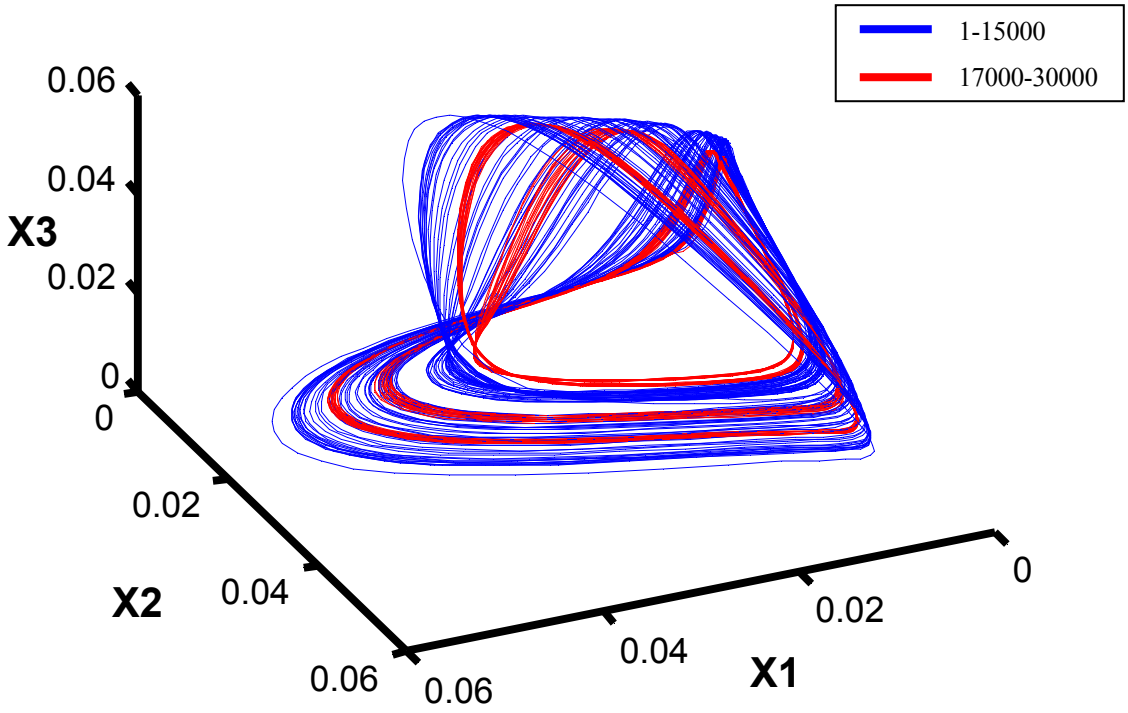


Figure 4.17: *Attractors form by the two different parameters sets are embedded into each other.*

This explains why the neural net model was able to predict data points in the latter part of the time series sequence (data points 20000-30000) reasonably accurately. The situation underlines the inherent problems when using models to detect dynamic change. Has the parameter change been the other way around, the model performance would have deteriorated much quicker. The manner in which the time series sequence is structured now, however, favours the model fitness, which is why the model-based technique did not detect nonstationarity with as much confidence.

4.1.5 Using the Correlation Dimension to Detect Dynamic Change in the Autocatalytic Reactor

The correlation dimension (d_c) is a statistic that characterizes the topology of an attractor. When process parameters change, the geometrical shape of the system attractor also changes. This makes the correlation dimension (d_c) potentially an "ideal" statistic for detecting nonstationarity. Especially when using Judd's algorithm, as it provides details about the geometry of the attractor on different length scales.

The same embedding parameters ($d = 3$ and $l = 17$) are used to reconstruct the attractor from which the $d_c(\varepsilon_0)$- curves will be calculated. The first step in detecting nonstationarity is to calculate the $d_c(\varepsilon_0)$- curves for the first and second halve of the time series. If the $d_c(\varepsilon_0)$-curves differ from each other, with respect to either *the region they occupy on the* $d_c(\varepsilon_0)$-ε *chart* and/or their *shape*, the process is presumed nonstationary. The $d_c(\varepsilon_0)$-curves for the two halves of the time series is shown in ***Figure 4.18***.

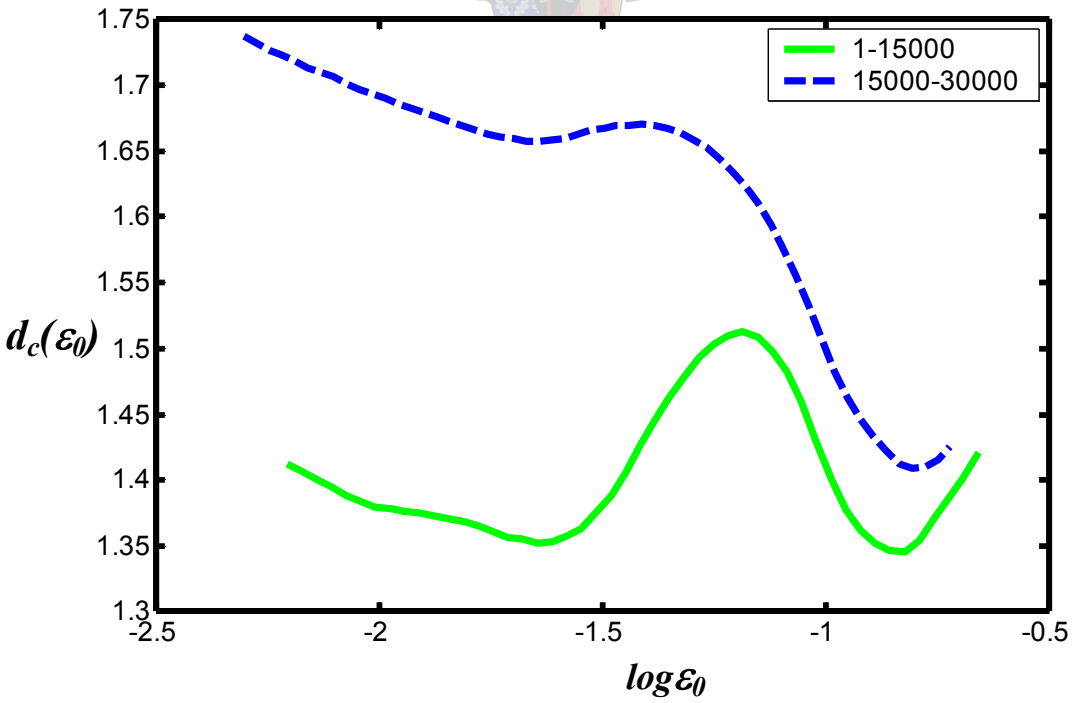


Figure 4.18: *$d_c(\varepsilon_0)$-curves from the two halves of the autocatalytic time series.*

The dissimilarity of the $d_c(\varepsilon_0)$-curves in ***Figure 4.18*** confirms nonstationarity. Although it has now been determined that the time series is nonstationary, the extent of the parameter change is still unknown. Dividing the data set into smaller segments (6 segments with 5000 points each) proves to be more useful (***Figure 4.19***).

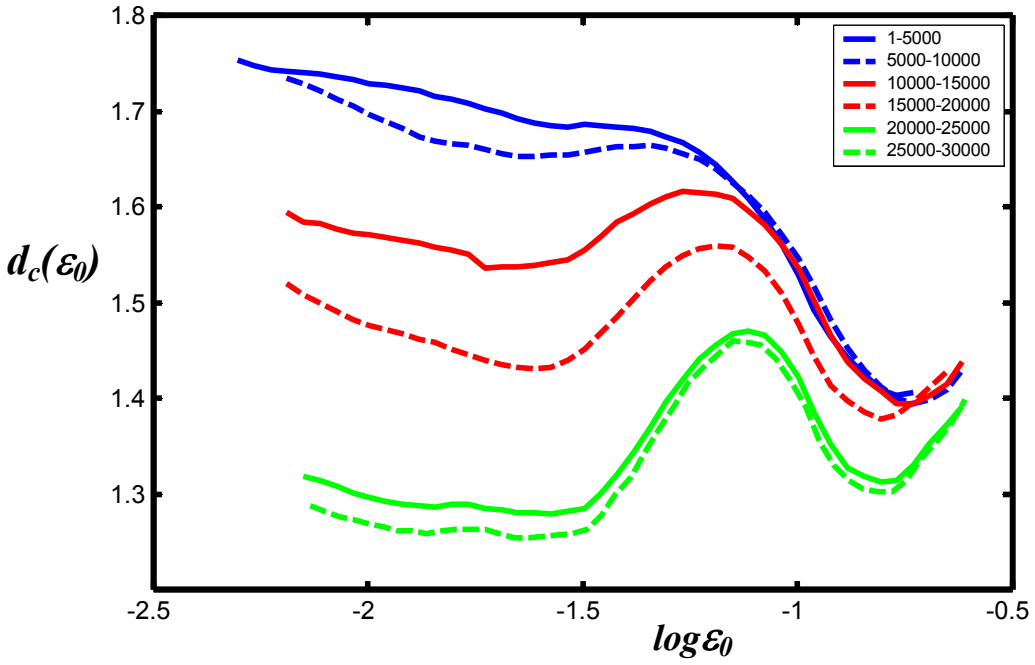


Figure 4.19: *$d_c(\varepsilon_0)$-curves from six segments, each containing 5000 data points, of the autocatalytic time series.*

The similarity of the first two $d_c(\varepsilon_0)$-curves (segments containing data points 1-5000 and 5000-10000) point towards the fact that no, or very slight, parameter change took place. The part of the time series containing the first 10000 data points is thus considered stationary. The parameter change is revealed by the $d_c(\varepsilon_0)$-curve from the third segment (10000-15000). The fact that this $d_c(\varepsilon_0)$-curve, as well as the curve from segment four (15000-20000), moves out of the "stationary" region, indicates a change in process parameters. The curves from the last two segments (20000-25000 and 25000-30000) are again very similar and suggest that the drifting parameters have stabilized at new values.

When analysing this particular graph (***Figure 4.19)*** it is clear that nonstationarity can only be reliably detected after observing 15000 data points from the process; although the parameter drift started just after the 10000th point. This can be improved by increasing the number of segments, and thus pinpointing the time at which the parameter change took place with greater accuracy. The only problem with this advance is that Judd's algorithm is very "data hungry" and gets less accurate with a decreasing number of data points. Furthermore, by decreasing the number of data points in a segment, one risks working with a data set that is nonstationary as a result of too few data points.

One way to avoid this problem is to use a moving window. ***Figure 4.20*** illustrates this approach using a fixed-size moving window of 6000 data points, which moves forward by 2000 data points with each step.

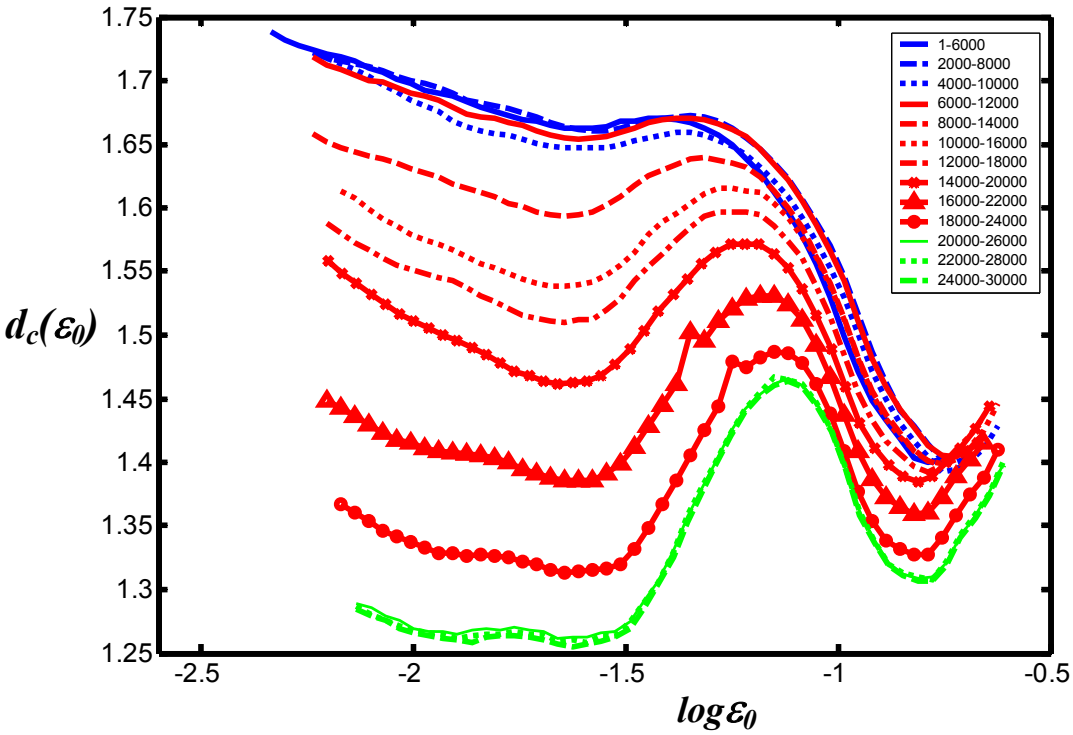


Figure 4.20: *Calculating $d_c(\varepsilon_0)$-curves from a moving window for the autocatalytic reaction time series.*

In this plot the blue curves represents the part of the time series were the process parameters are constant, the red curves where the parameters are in the process of changing, and the green curves the part where the parameters stay fixed at their new value. No parameter change can be detected from the first four moving window

segments (up to data point 12000). The fifth moving window segment (8000-14000) shows the first signs of a possible parameter change. The moving window approach is an improvement on the previous 6-segment approach in that only 14000 data points are necessary to detect the parameter drift, compared to 15000. The fourth moving window segment (6000-12000) is not capable of picking up the change, as 4000 of the 6000 data points are still part of the stationary part of the time series and only slight parameter change takes place in the next 2000 points.

The moving window approach can be especially useful for online monitoring of changes in process parameters, but also where the available amount of data is limited.

4.1.6 The Effect of Noise in the Autocatalytic Process Data

One might argue that data measured from real-life processes are not, as is the case for the previous simulated data, noise free. To test the effectiveness of the change detection techniques on noisy data, Gaussian measurement noise was added to the autocatalytic process data used in the previous case study. The noise level was set to 10% of the standard deviation of the data. The effect of adding measurement noise to the data is evident from ***Figure 4.21***.

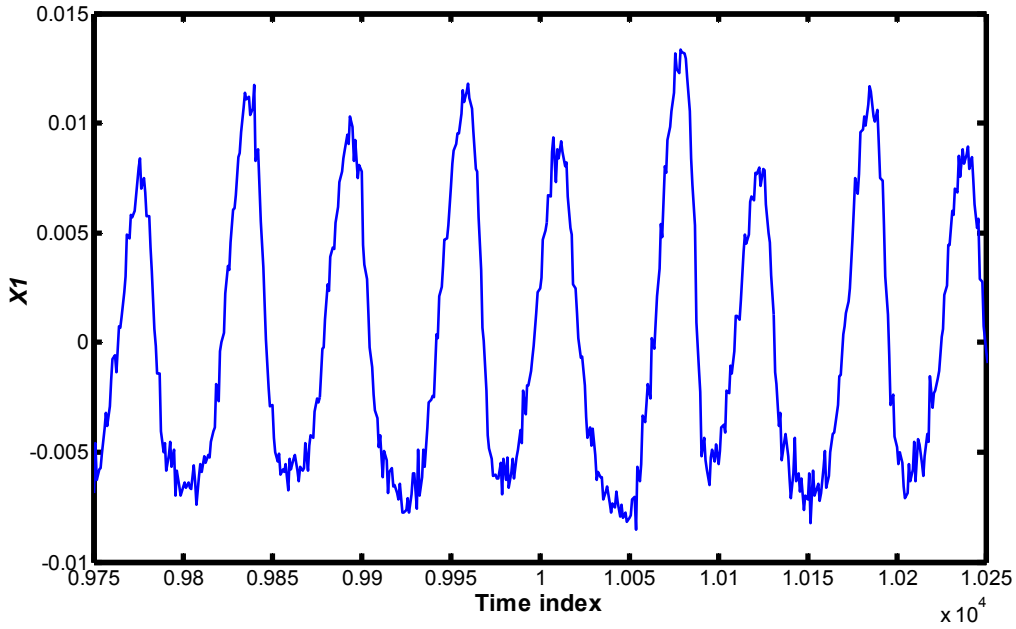


Figure 4.21: *Data points 9750 to 10250 from the nonstationary time series generated by the autocatalytic reaction – noisy data.*

As in the case for the noise free data, the optimal embedding parameters are determined by the AMI and FNN statistics. An embedding delay of $l = 15$ (***Figure 4.22***) and an embedding dimension of $d = 6$ (***Figure 4.23***) are chosen. While the embedding delay for the noisy data is close to that of the noise free data (15 time steps vs. 17 time steps), the embedding dimension for the noisy data are considerably higher compared to that of the noise free data (6 dimensions vs. 3 dimensions). This demonstrates that noise increases the dimensionality of the data. The resulting reconstructed attractor is shown in ***Figure 4.24***. The presence of noise in the reconstructed attractor is clearly visible; as well as the dissimilarity in shape when compared to that of the noise free data.

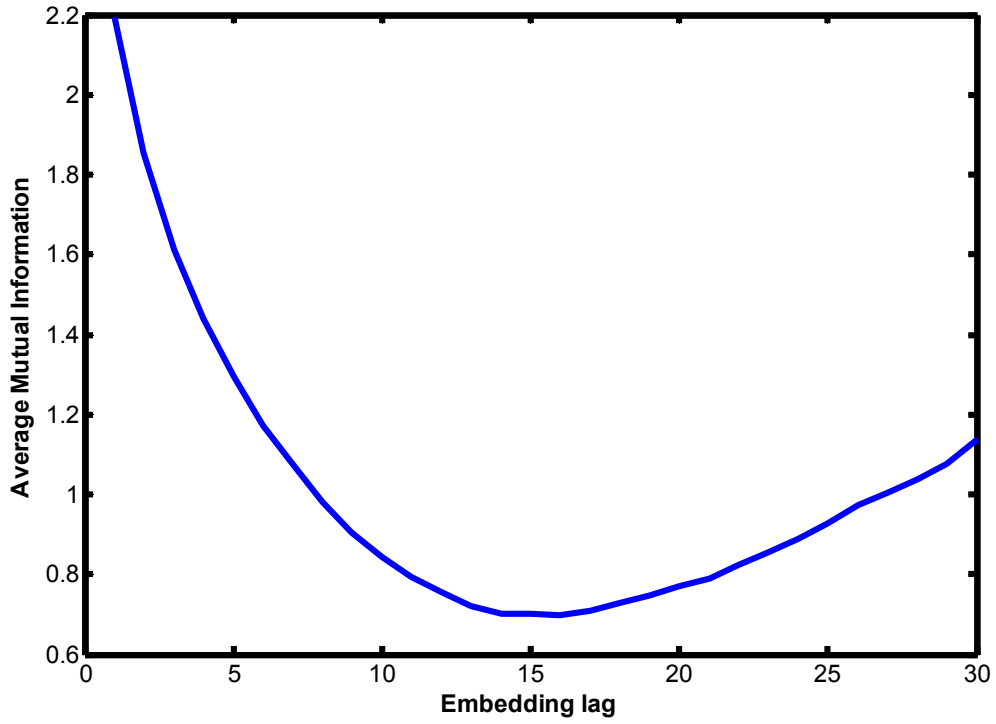


Figure 4.22: *The average mutual information, as function of the time delay, for the autocatalytic reaction – noisy data.*

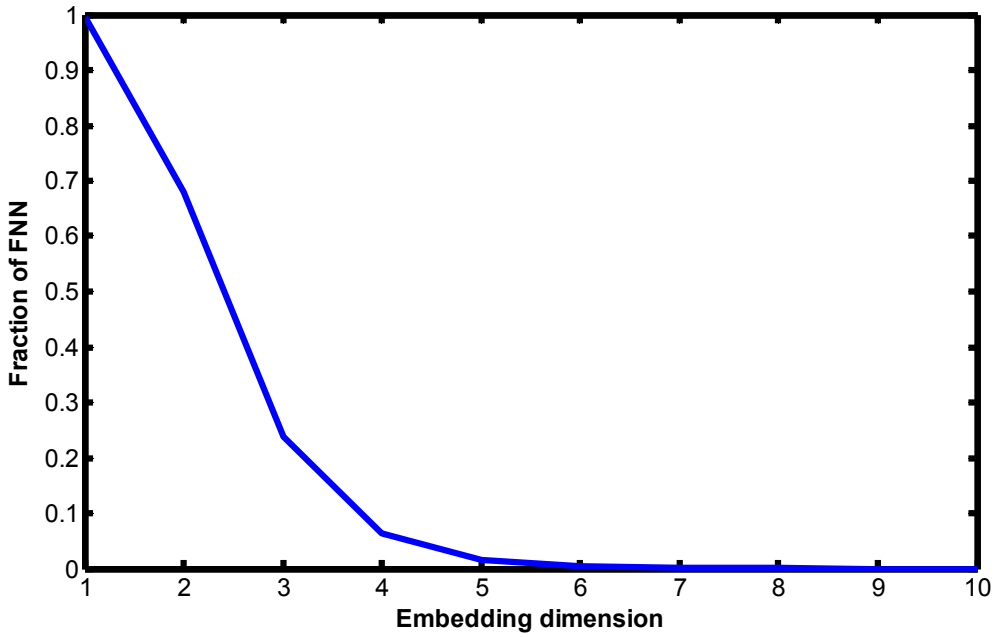


Figure 4.23: *Fraction of FNN as function of embedding dimension for the autocatalytic reaction – noisy data.*

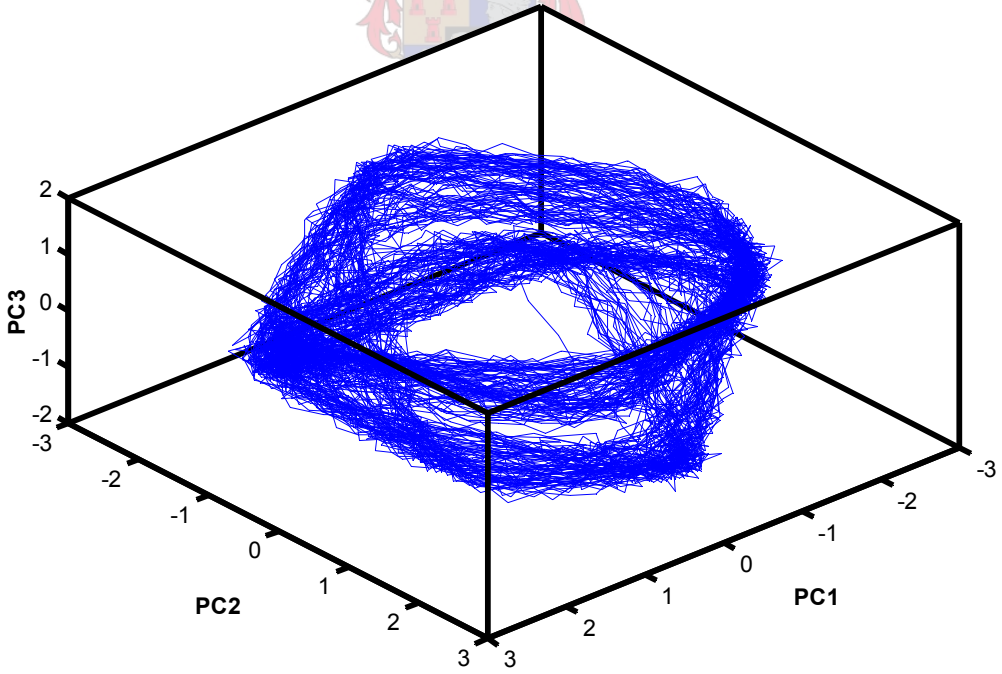


Figure 4.24: *Reconstructed attractor for the autocatalytic reaction system, projected onto the first three principal components – noisy data.*

Surrogate analysis of the noisy data also suggests that the process which generated the data could be nonlinear deterministic. The noise had no effect on the analysis, apart from slightly increasing the $d_c(\varepsilon_0)$-values for the actual data.

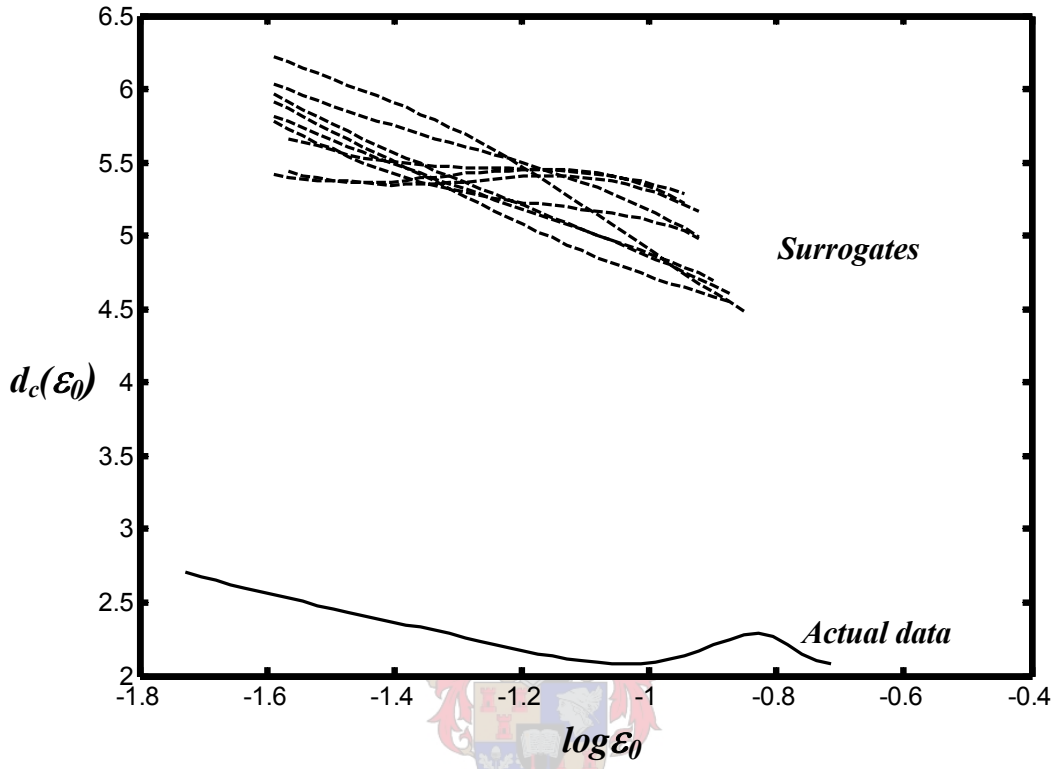


Figure 4.25: *Correlation dimension curves for surrogates and actual data for autocatalytic reaction – noisy data.*

The same approach, used for the noise free data, is used to detect dynamic changes in the noisy data. The model parameters are estimated from the first 5000 data points of the noisy time series, and then used to predict *unseen* data from the rest of the time series. ***Figure 2.26*** shows a free-run prediction of the model on the first stationary part of the time series. The performance of the model deteriorates when predicting data from the last part of the time series, which suggest that there was a change in the process parameters (***Figure 2.27***). This is consistent with the results obtained from the noise free data.

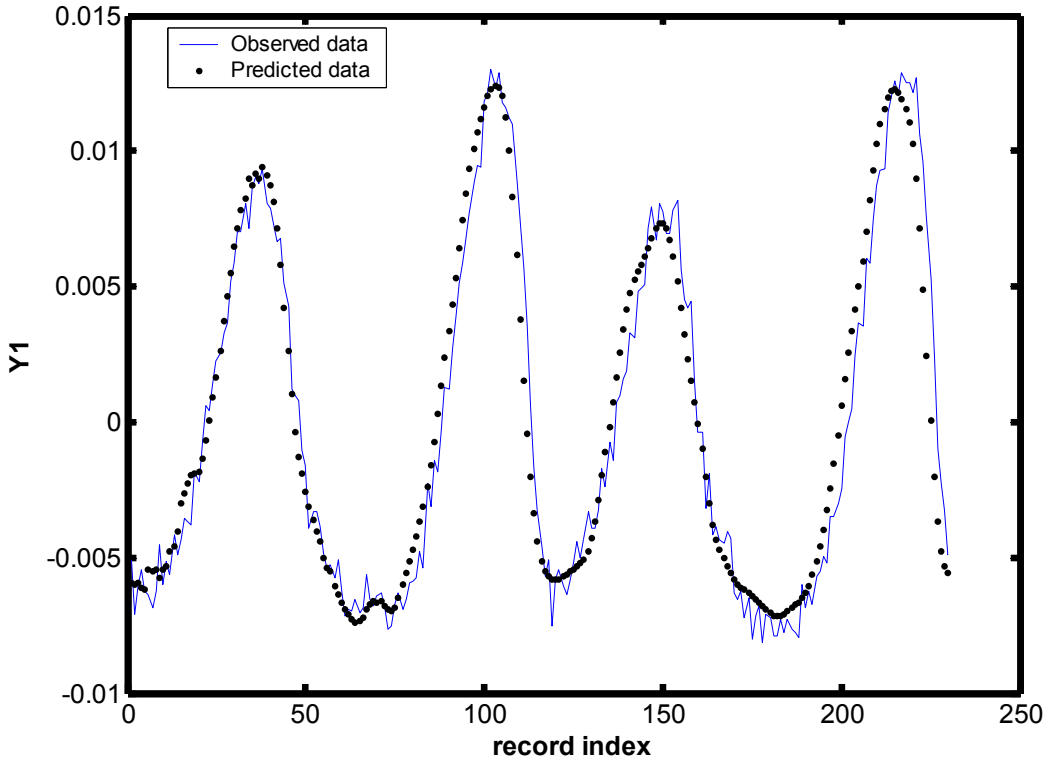


Figure 4.26: *Free run prediction on the first part of the time series where the process parameters were unchanged – noisy data*

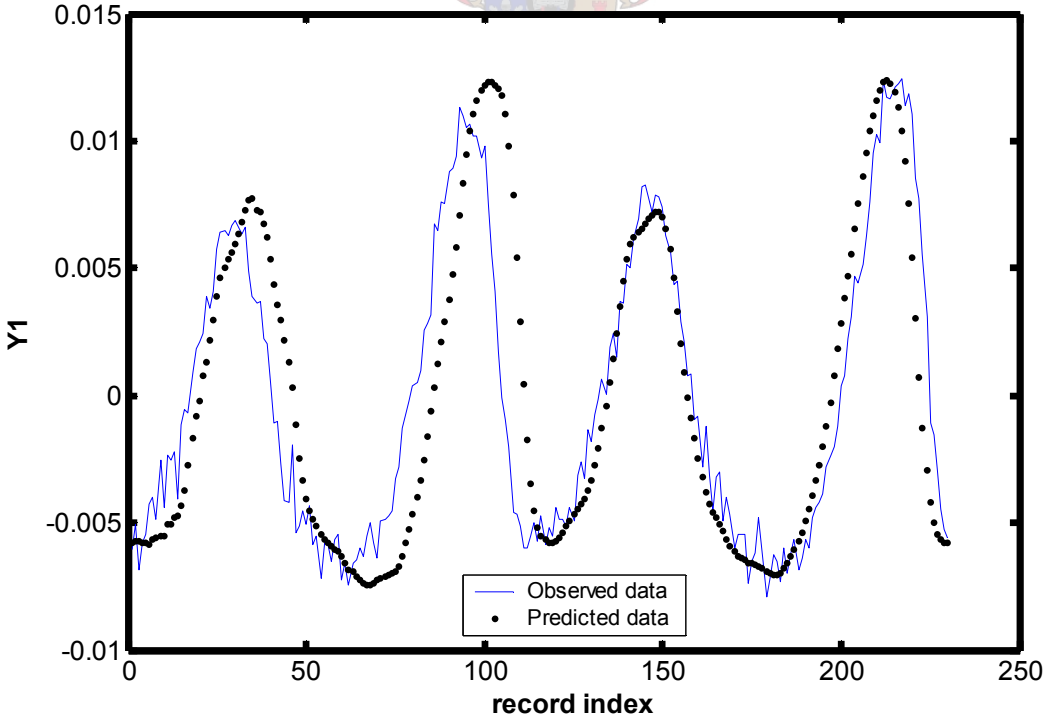


Figure 4.27: *Free run prediction on the last part of the time series where the process parameters had already changed – noisy data.*

The method of nonlinear cross predictions (Schreiber, 1997) gives essentially the same results (a parameter change roundabout data point 15000) on the noisy data, as it did on the noise free data (***Figure 4.28*** and ***Figure 4.29***). The only real difference is the magnitude of the prediction errors. As one would expect, the prediction errors are larger for the noisy data than for the noise free data.

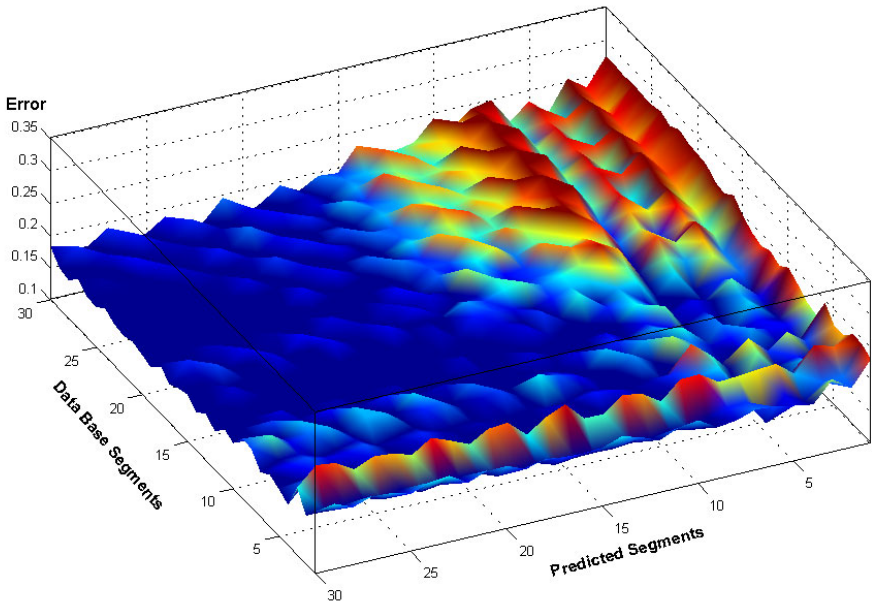


Figure 4.28: *3-D surface plot of mutual cross-prediction errors for the autocatalytic reaction – noisy data.*

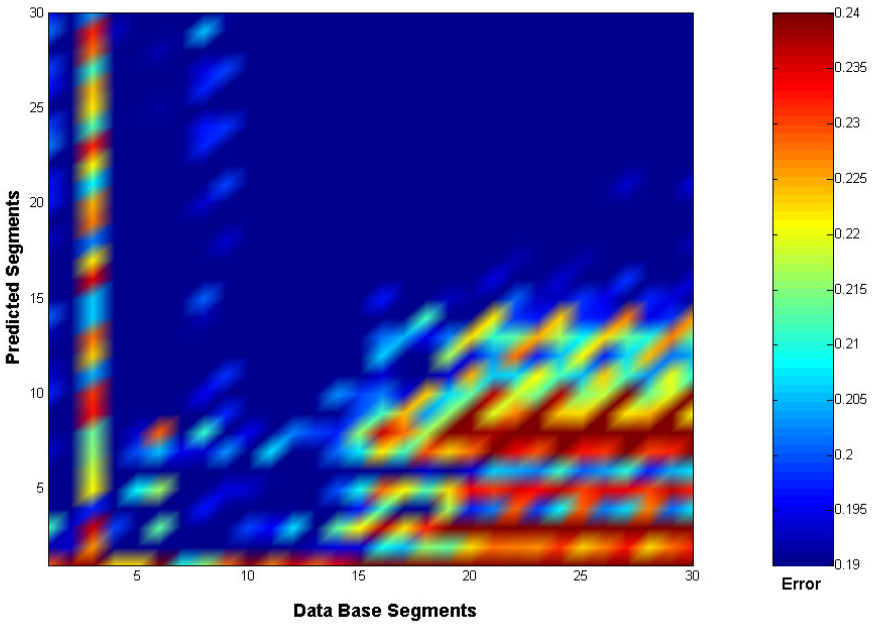


Figure 4.29: *2-D colour-coded mutual prediction map for autocatalytic reaction – noisy data.*

An assessment of the correlation dimension statistics from the noisy data gives similar change detection results to that obtained from the noise free data. The moving window approach (***Figure 4.30***) confirms a parameter change from the 5th moving window segment (data points 8000-14000) onwards.

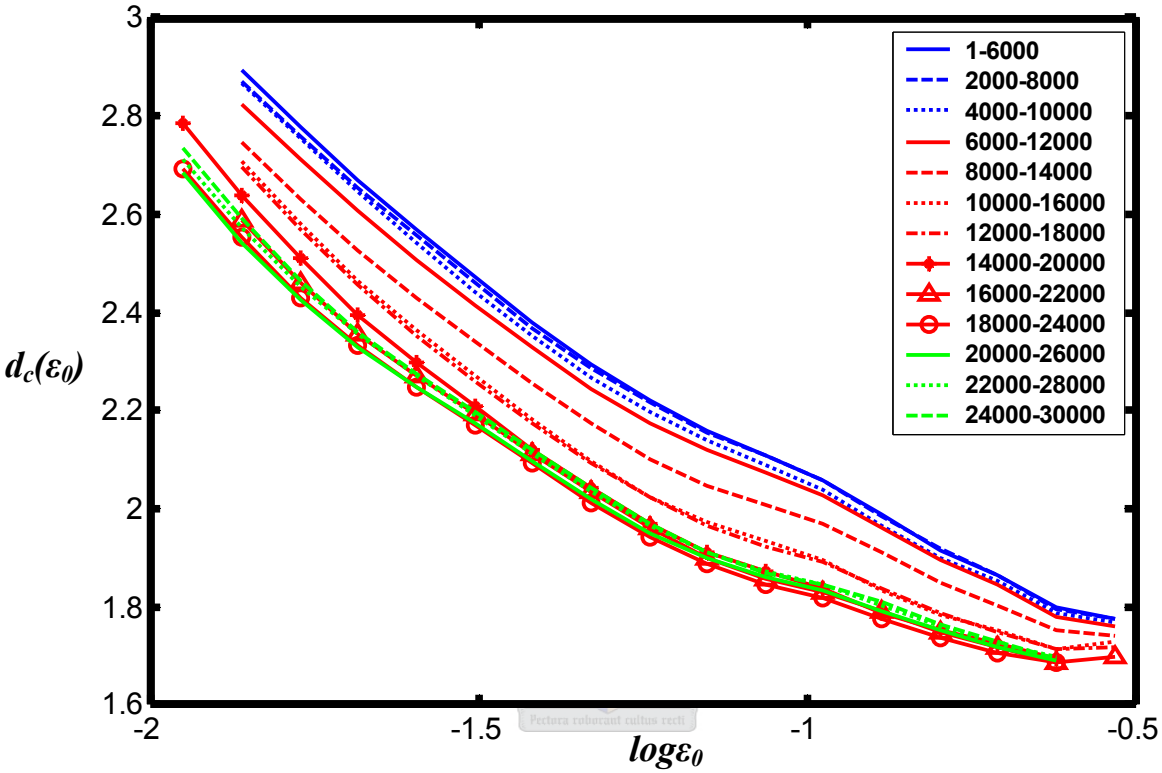


Figure 4.30: *Calculating $d_c(\varepsilon_0)$-curves from a moving window for the autocatalytic reaction time series – noisy data.*

The amount of measurement noise added to the data in this case study, did not have a significant impact on the ability to reliably detect changes.

4.2 The Baker's Map

Although this system does not represent a chemical process, it is a well-known chaotic system and is a popular choice for *benchmarking* stationarity tests (Schreiber, 1997; Yu et al., 1998; Yu et al., 1999). This is the main motivation for using the baker's map as a case study, as it will give a good idea of how the change detection techniques discussed in this work perform in comparison to other proposed methods.

A generalization of the baker's map is given by the equations,

$$\begin{aligned} (v_n \le \alpha) \Rightarrow & \begin{matrix} u_{n+1} = \beta u_n \\ v_{n+1} = v_n / \alpha \end{matrix} \\ (v_n > \alpha) \Rightarrow & \begin{matrix} u_{n+1} = 0.5 + \beta u_n \\ v_{n+1} = (v_n - \alpha)/(1-\alpha) \end{matrix} \end{aligned} \quad . \tag{4.14}$$

The system is a two dimensional piecewise linear mapping defined for $v_n \in [0,1]$, α and $\beta \in]0,1[$. A nonstationary time series is created by varying β slowly with time,

$$\beta = \frac{n}{N},$$

while the parameter α is kept fixed at $\alpha = 0.4$. Forty thousand data points are measured ($N = 40000$) by recording $w = u + v$. The running mean is subtracted from this value (w_n), and the time series is normalized to unit running variance. The resulting time series becomes:

$$x_n = \frac{w_n - \langle w \rangle_k}{\sqrt{\left\langle \left(w - \langle w \rangle_k \right)^2 \right\rangle_k}} \tag{4.15}$$

Here $\langle \cdot \rangle_k$ represents the average over the indices $n' = (n-k),...,(n+k)$, with $k = 50$ in this instance. Since most of the observables in this sequence remain unchanged, the nonstationarity is very hard to detect. The running mean and variance are constant up to finite sample fluctuations, and autocorrelations show only very small variation (Screiber, 1997). When inspecting the entire time series visually (***Figure 4.31(a)***), one would only suspect that the process parameters have changed after observing at least

25000 data points (indicated by an increase in the variance of the observations). Since the system is subjected to a slow parameter drift, chances of observing parameter changes when viewing only a small number of data points are even more remote (***Figure 2.31(b)***).

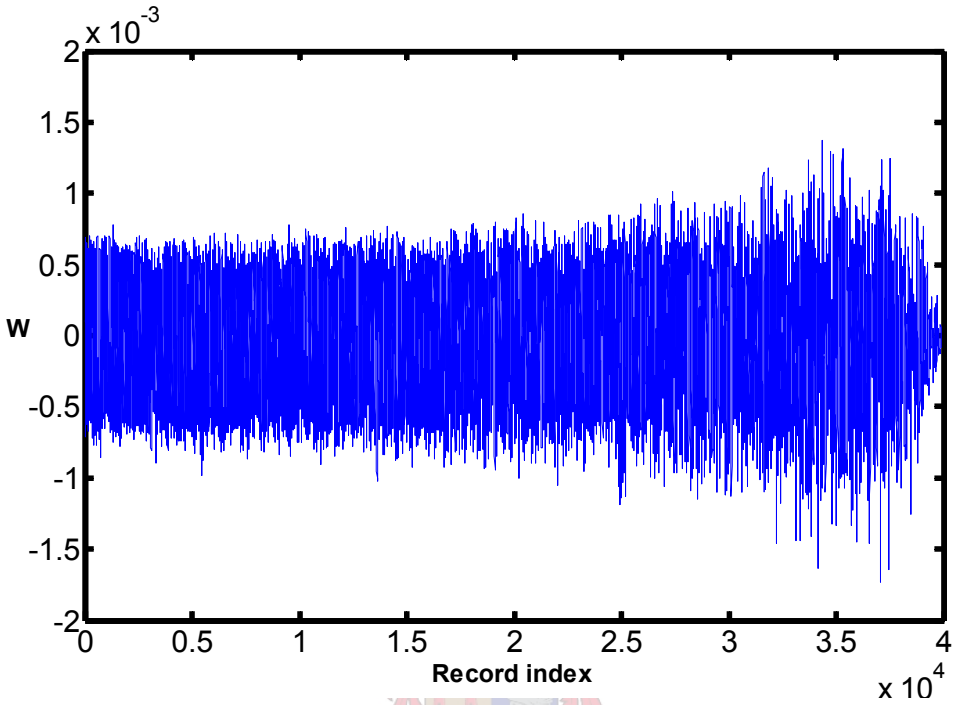


Figure 4.31(a): *Time series data for the nonstationary baker's map – all 40000 data points*

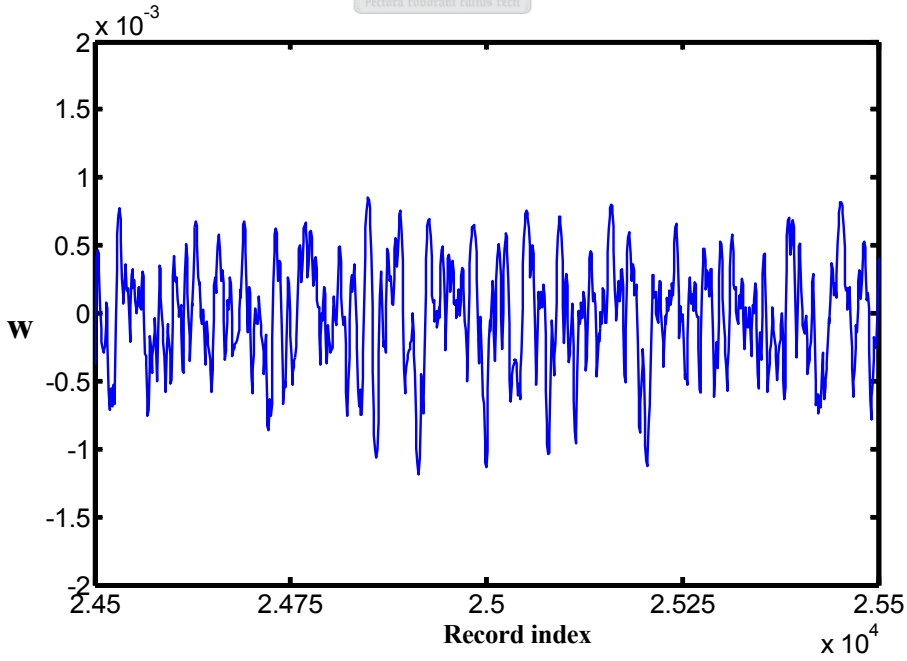


Figure 4.31(b): *Time series data for the nonstationary baker's map – data points 24500 to 25500.*

4.2.1 State Space Reconstruction of the Baker's Map

Researchers that used the baker's map for benchmarking their stationarity tests, reconstructed the attractor by embedding the time series into two dimensions ($d = 2$) using a time delay of one ($l = 1$). To be consistent and ensure comparable results, the same embedding parameters are used in this case study. The reconstructed attractor is shown in ***Figure 4.32***.

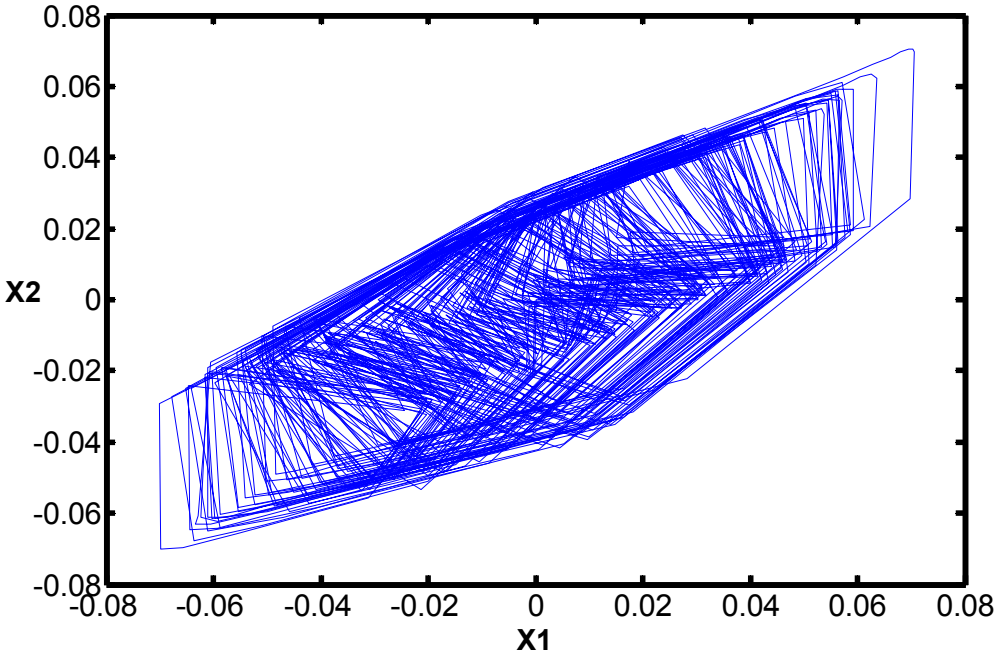


Figure 4.32: *Reconstructed attractor for the baker's map.*

4.2.2 Surrogate Data Analysis for the Baker's Map

To determine if the baker's map exhibit nonlinear deterministic behaviour, 15 *type 2* surrogate data sets are generated, again using the AAFT algorithm. The $d_c(\varepsilon_0)$-curves from the actual data and surrogate data (embedded in 10 dimensions) are compared in ***Figure 4.33***. The result from the surrogate data analysis cannot confirm nonlinear determinism with confidence. While the $d_c(\varepsilon_0)$-curve for the actual data is lower dimensional and separated from the surrogates for $\log \varepsilon_0 < -1.6$, the opposite is true for the larger length scales. This inconclusive result is most probably due to instabilities in the correlation dimension algorithm.

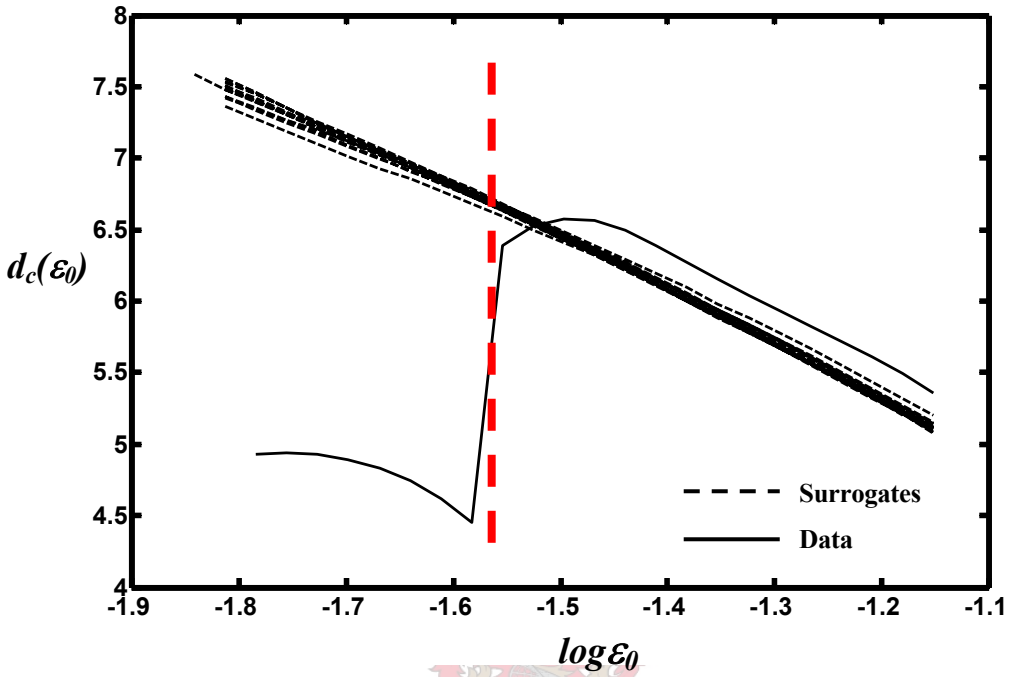


***Figure 4.33**: $d_c(\varepsilon_0)$-curves for surrogate and actual data when embedding into 10 dimensions.*

4.2.3 Modelling the Baker's Map

Modelling the baker's map proved to be more challenging than that of the autocatalytic system. In the case of the autocatalytic system, a one-step prediction R^2-value of essentially *one* were obtained on "unseen" data, but the best model for the baker's map could only achieve a R^2-value of 0.912. The optimal model, as determined by the SIC, has 11 hidden nodes (***Figure 4.34***).

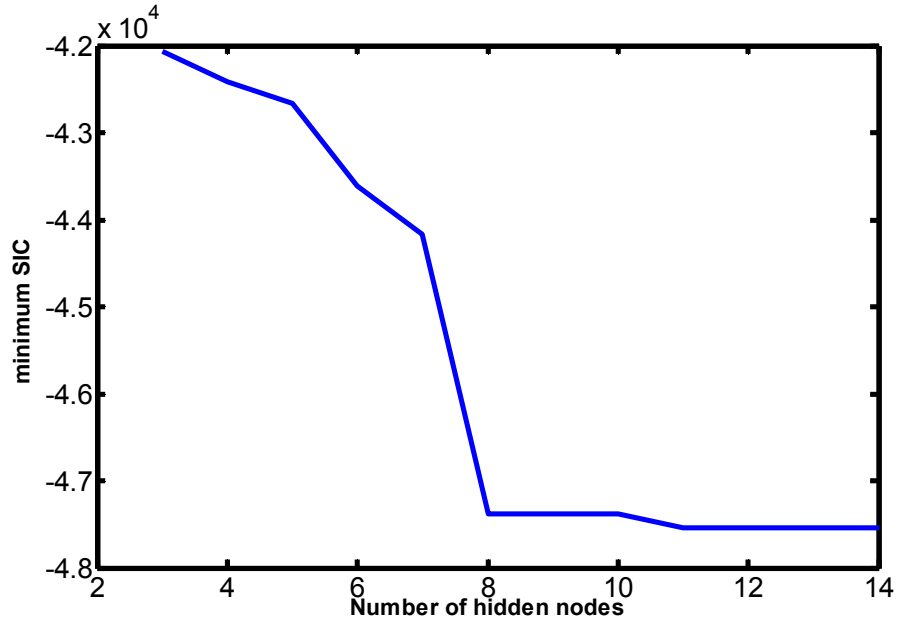


Figure 4.34: *History of the moving global minimum of the Schwartz Information Criterion versus the number of hidden nodes for the baker's map.*

Since the estimated model does not provide a very good fit to the data, the one step prediction, rather than the free-run prediction, will be used to validate the model. It will be useless to compare free-run predictions for detecting nonstationarity, as even the training ("seen") data gave bad free-run results (***Figure 4.35***).

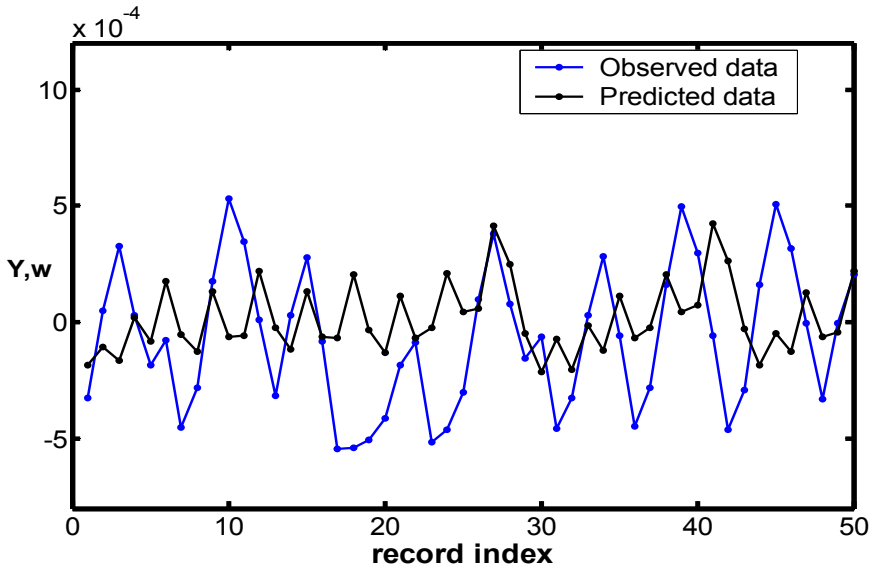


Figure 4.35: *Free-run prediction of model training data for baker's map.*

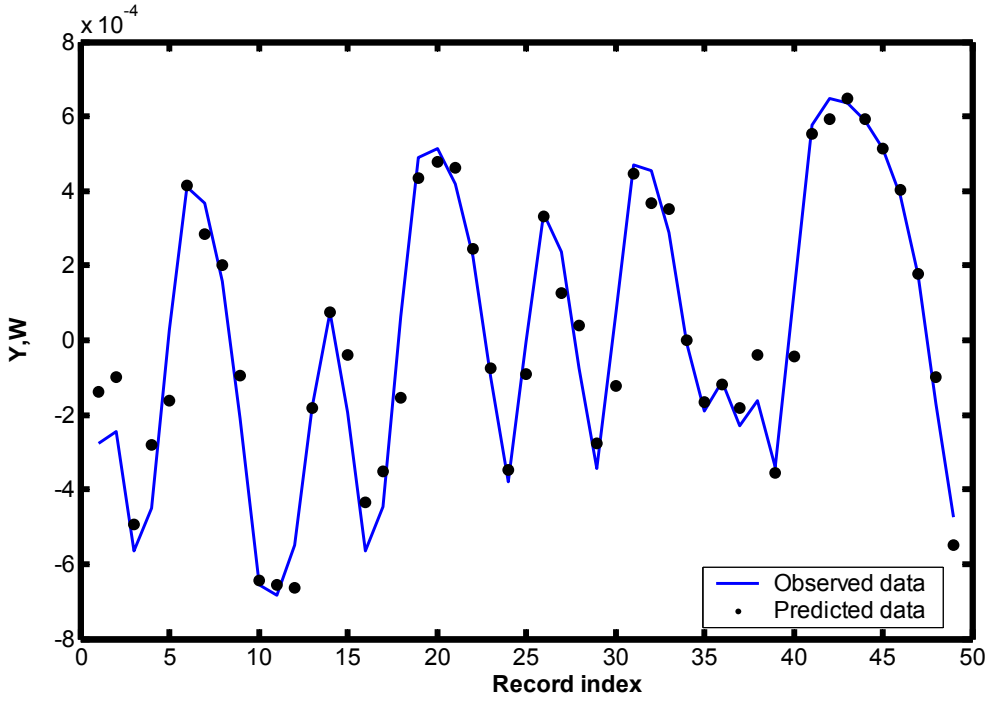


Figure 4.36: *One step prediction for data points 7700 to 7750 to illustrate the model fitness.*

The R^2 validation results, as calculated from different segments of the time series, are given in ***Table 4.1. Figure 4.36*** shows the one step prediction model fit for the segment 5000-10000.

Table 4.1. *R^2 - statistic results for one-step prediction of different segments from the baker's map time series.*

Data Segment	***R^2-statistic***	***Data Segment***	***R^2-statistic***
5000-10000	0.857	25000-30000	0.68
10000-15000	0.792	30000-35000	0.73
15000-20000	0.675	35000-40000	0.761
20000-25000	0.622		

It is hard to make meaningful conclusions, with respect to detecting parameter change, when validating the model on consecutive segments of the time series. At first, it seems as if the model gets worse when predicting observations further away from the training data set. This is evident when comparing the decreasing R^2-values for segments 5000-10000 to 20000-25000. For segments 25000-30000 to 35000-40000 the R^2-values increase again, which could lead one to believe that the parameters must be returning to their previous states. This is not the case.

The fact that the R^2-values from different segments of the time series is not statistically the same, suggests that the process generating the data might be nonstationary. Although this approach at least suggests that there might be parameter change, it gives no real indication of the extent of the nonstationarity.

4.2.4 Detecting Dynamic Change in the Baker's Map Using Nonlinear Cross Prediction

For this approach, the baker's map time series is divided into 40 segments of 1000 data points each, and embedded into two dimensions with a time delay of one. The mutual nonlinear cross predictions between the segments are again illustrated by means of a 3-D surface plot (***Figure 4.37***) and a 2-D colour-coded surface (***Figure 4.38***).

Both figures indicate that predictability degrades rapidly with the temporal distance of the segments. Nonstationarity can only be confirmed with confidence after taking into account at least 15 to 20 segments. In other words, about 15000 to 20000 data points must be observed before a change in the process parameters can be detected. The fairly constant rate (evident when analysing ***Figure 4.37***) at which the predictability degrades indicates that the change in dynamics may be as result of a slow parameter drift, rather than to an instantaneous parameter change.

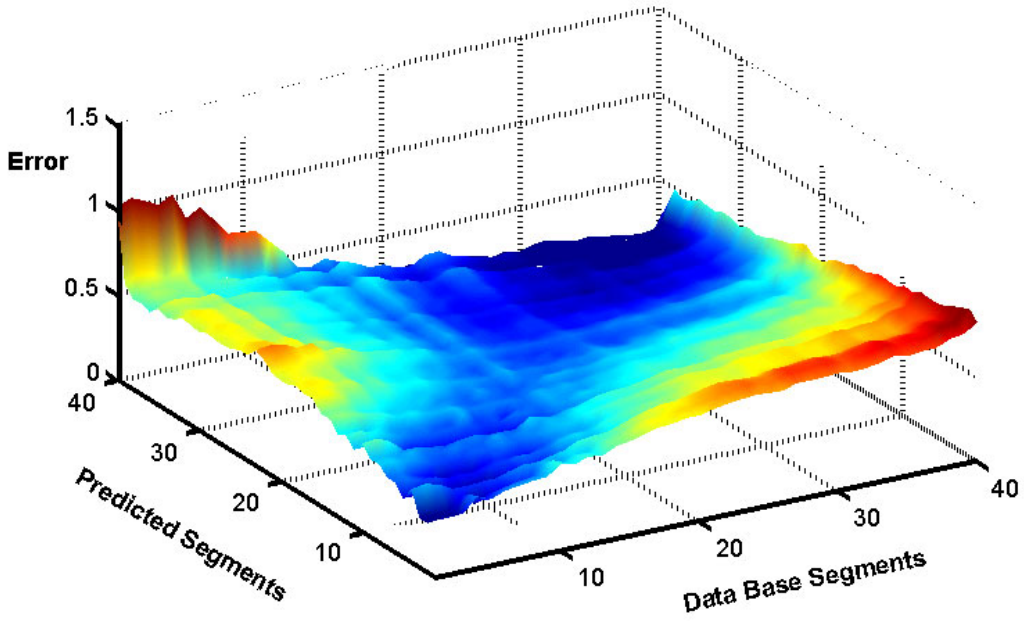


Figure 4.37: *3-D surface plot of mutual cross-prediction errors for the baker's map*

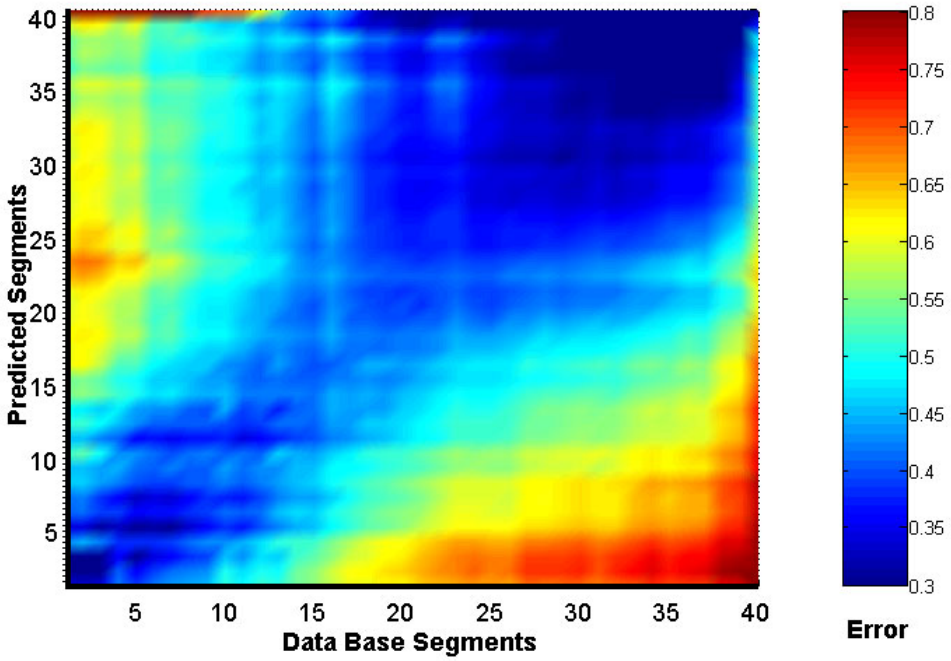


Figure 4.38: *2-D colour-coded mutual prediction map for the baker's map*

4.2.5 Using the Correlation Dimension to Detect Dynamic Change in the Baker's Map

Using the same embedding parameters than before ($d = 2$ and $l = 1$), a $d_c(\varepsilon_0)$-curve for each of the two halves of the baker's map time series is calculated as a starting point. The dissimilarity of the two curves (***Figure 4.39***) indicates that there must be differences in the geometrical structure of the attractors reconstructed from the two halves of the time series – suggesting nonstationarity in the data.

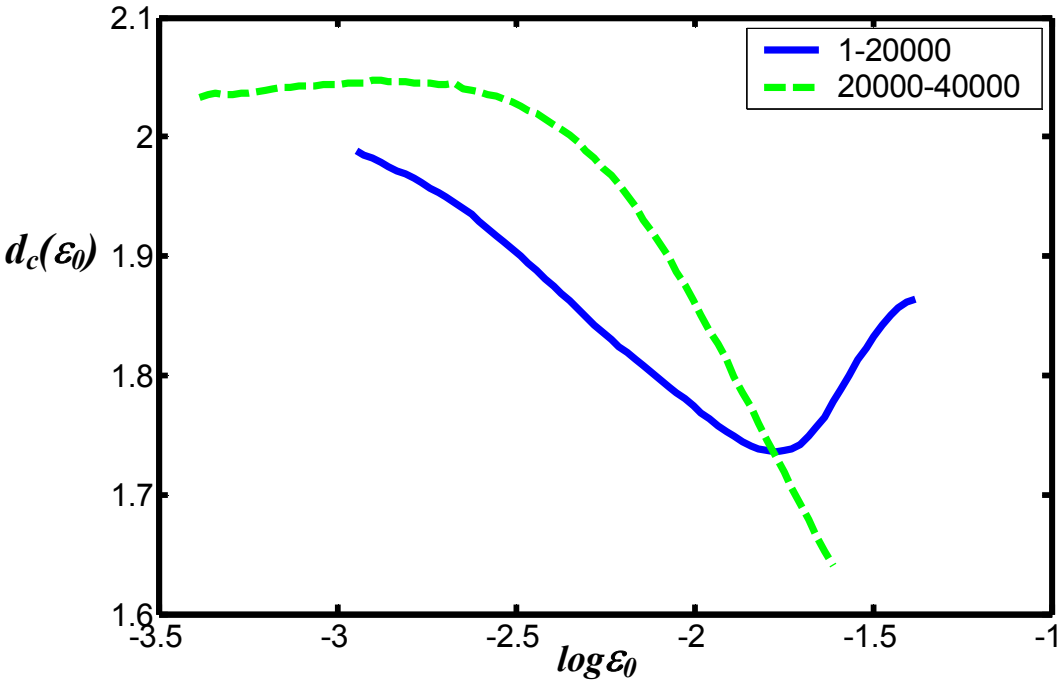


Figure 4.39: *$d_c(\varepsilon_0)$-curves from the two halves of the baker's map time series.*

To get a better idea of when the parameter change took place, and the extent of the change, the time series is divided into 8 segments containing 5000 data points each. The $d_c(\varepsilon_0)$-curves calculated from these segments are given in ***Figure 4.40***. Dissimilarity in the curves calculated from the consecutive segments of the data again confirms that the process parameters have not stayed constant throughout the observation period. No two curves look similar, which implies that there is a continuous parameter drift

throughout the observation period. Using this particular segmentation, it is not possible to identify any stationary part in the time series.

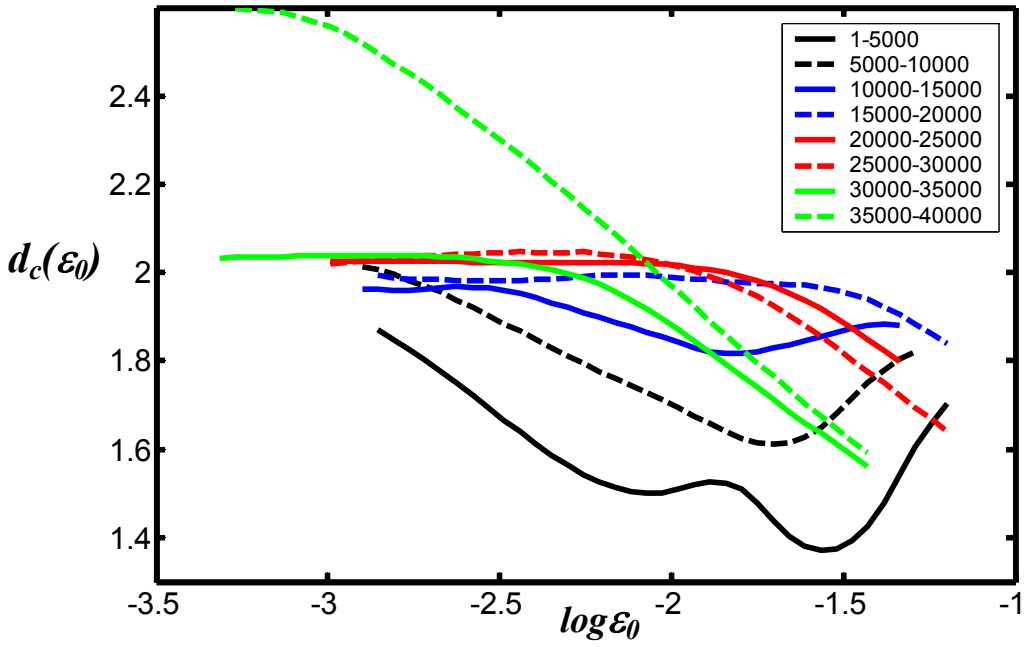


Figure 4.40: *$d_c(\varepsilon_0)$-curves calculated from eight consecutive segments, each containing 5000 data points, of the baker's map time series*

It would be interesting to see how quickly nonstationary can be confirmed. In other words, how many data points are necessary before a change in the parameters can be detected with confidence? The moving window approach is better suited for such an analysis. Using a window size of 5000 data points with a step size of 1000 data points generates ***Figure 4.41***. A comparison of the first two curves from the moving window segments already suggests that nonstationarity may be present. Comparing curves from consecutive moving window segments confirms the suspicion of nonstationarity in the data. Only about 7000-8000 observations are necessary to reliably detect nonstationarity, using correlation dimension as test statistic, in the moving window approach.

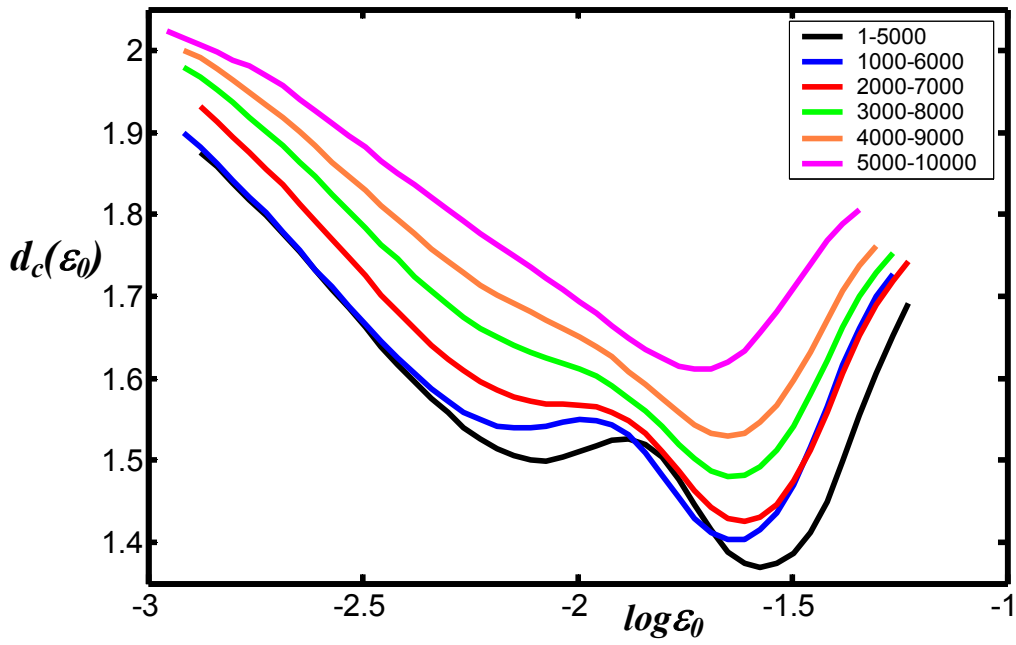


Figure 4.41: *Calculating $d_c(\varepsilon_0)$-curves from a 5000 point moving window for the baker's map.*

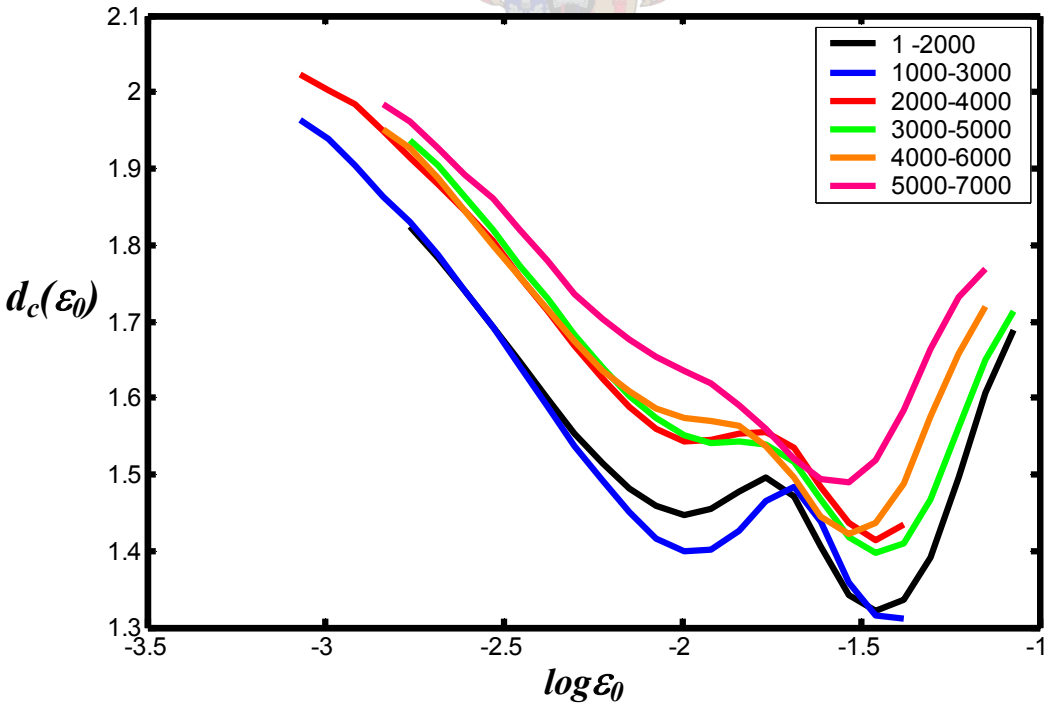


Figure 4.42: *Calculating $d_c(\varepsilon_0)$-curves from a 2000 point moving window for the baker's map.*

The size of the moving window can even be reduced to 2000 data points, to reduce the number of data points needed to detect the parameter change (***Figure 4.42***). One should just remember that decreasing the window size, increases the risk of working with segments that are nonstationary as result of too few data points. In this instance, only 4000 data points is necessary to reliably detect nonstationarity. This is an improvement on Schreiber's mutual nonlinear cross prediction method where at least 15000 observations are needed.

It also outperforms the technique used by Yu et al. (1998 & 1999) where 5000 to 10000 observations are needed to reliably detect such hidden nonstationarity. They make use of *space time-index plots* to probe dynamical nonstationarities in the data. They believe that in a space time-index plot, the density distributions as a function of normalized time-index are V-shaped when the data are nonstationarity.

4.3 Real Data from a Metal Leaching Plant

The problem with using actual plant data to evaluate dynamical change detection techniques, is to obtain time series data that are believed (or known) to be nonstationary. One way to do this is by visual inspection, although it can sometimes be misleading. In this case study, data from a metal leaching plant will be used (***Figure 4.43***). The data set consists of 1747 observations, where the concentration of the valuable metal is the measured variable. The time interval between successive observations is one day. Although the time span of the data set under consideration may seem long (about 5 years), a dynamic change in this process can typically be as result of a change in the composition of the ore. Such a change may take years to manifest. So even if the change detection techniques are only able to detect changes in the process after say a few years, it is still fast compared to the dynamics of the overall process.

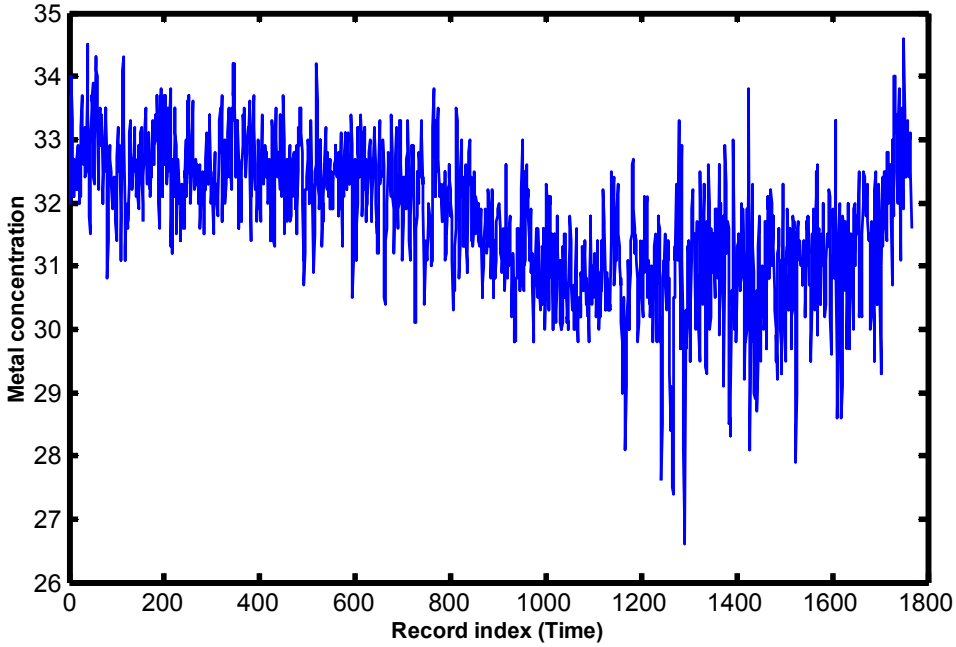


Figure 4.43: *Data from a metal leaching plant.*

Visual inspection of the data plot suggests that there might be a change in the process parameters roundabout observation 800. The data prior to this point look more or less stationary, but the dynamic behaviour of the process from that point onwards seems to have changed. To remove the obvious linear nonstationarity (changing mean) from the data, a 10-point moving average is subtracted from the observations. The resulting time

series is shown in ***Figure 4.44***. The *potential* nonstationarity in the data is now less obvious.

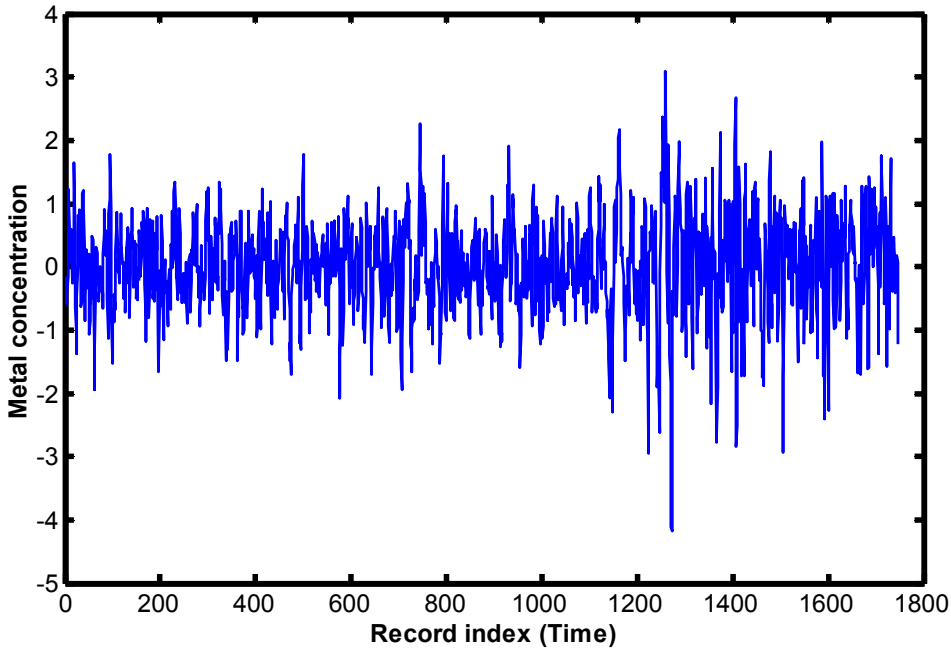


Figure 4.44: *Linearly adjusted data from the metal leaching plant*

The fact that the data set is very short might prove to be problematic. When evaluating measures for detecting nonstationarity, large data sets are preferred. Results become increasingly more unreliable with shorter data sets.

4.3.1 State Space Reconstruction of Metal Leaching Data

Being a real-life process, one must assume that there is some degree of measurement noise present in the data. State space reconstruction by means of singular spectral analysis (SSA) is more robust against noisy data when compared to traditional embedding methods. One of the reasons is that the PCA step of SSA filters out some of the unwanted noise, which normally results in a "cleaner" reconstructed attractor (Kugiumtzis & Christopherson, 1997). It is for this reason that SSA will be used to reconstruct the attractor.

The plot of the autocorrelation function suggests an initial embedding dimension of 5 (***Figure 4.45***). The eigenvalues of the covariance matrix are given in ***Figure 4.46***. Each of the eigenvalues explains a certain variance in the data. When the initial embedding dimension for a given data set is large (e.g. > 10), the embedding dimension can be reduced by selecting only the most prominent eigenvalues, i.e. the eigenvalues that explain most of the variance. The embedding dimension is then reduced to the number of eigenvalues chosen and PCA is used to reduce correlations between the reconstructed vectors.

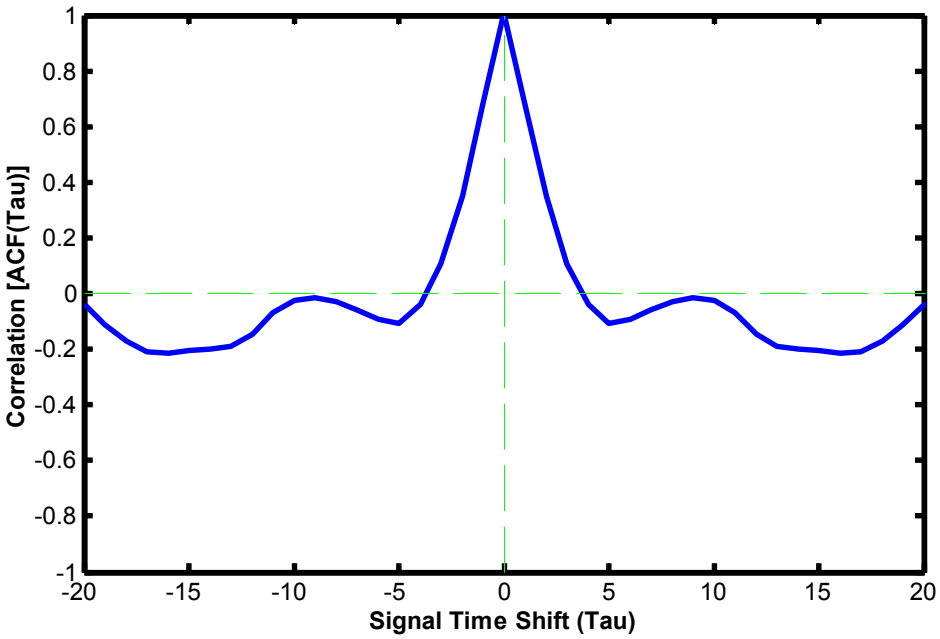


Figure 4.45: *Autocorrelation function of the metal leaching data*

It is important however to make sure that at least 95% (rule of thumb) of the variance in the data can still be explained by the retained eigenvalues. In this case, there is no need to reduce the dimension as the initial embedding dimension is already at a low value of 5. The reconstructed attractor (***Figure 4.47***) does not seem to have any geometrical structure, as is expected from a deterministic system. This does not automatically mean that the data are stochastic. Remember that the attractor can only be viewed in three dimensions (two dimensions actually, as it is viewed on a flat surface), although the actual reconstructed attractor is 5-dimensional. In three dimensions, the attractor is

projected onto its three principal axes as determined by the three largest eigenvalues. The first three principal components explain only about 91.6 % of the variance in the data. It is therefore still possible that the attractor has some geometrical structure in five dimensions. The best way to see if the data exhibit some deterministic behaviour is by doing surrogate data analysis.

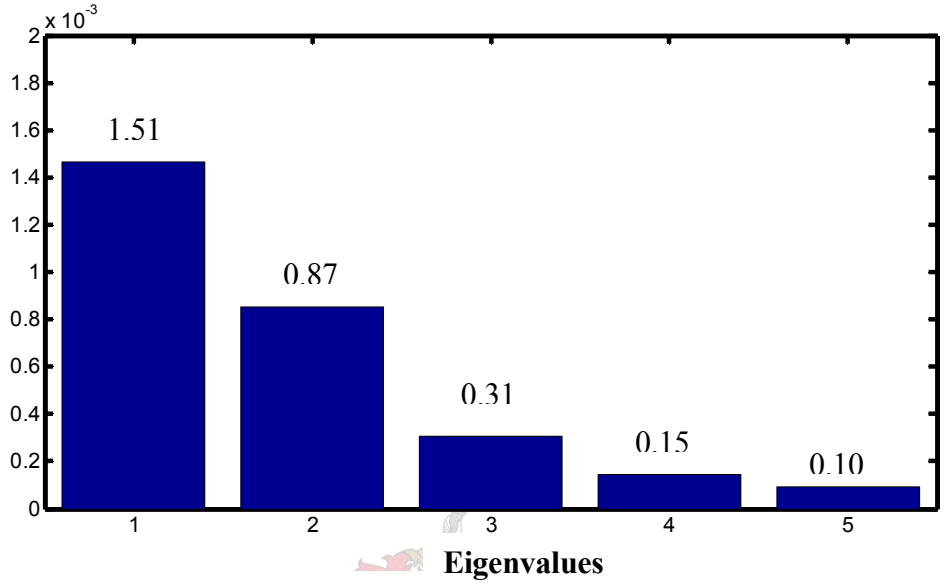


Figure 4.46: *Eigenvalues of the covariance matrix of the metal leaching data.*

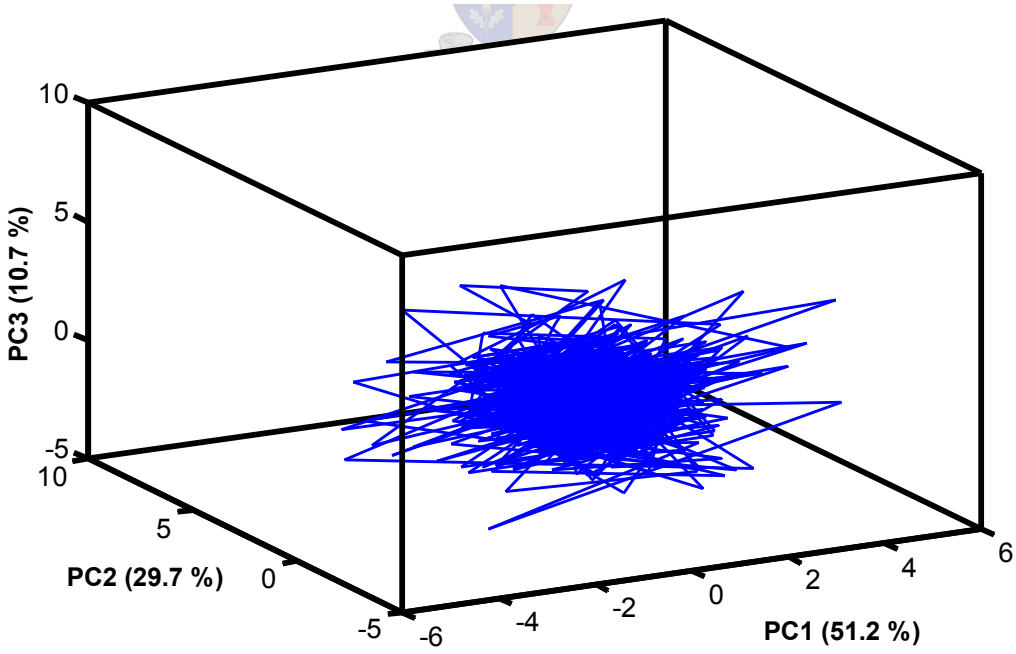


Figure 4.47: *Reconstructed attractor for the metal leaching data, projected onto the first three principal components. The variance explained by each principal component is given in brackets.*

4.3.2 Surrogate Data Analysis of the Metal Leaching Data.

The AAFT algorithm is used to generate 15 type 2 surrogate data sets, which is then compared to the actual data set using the correlation dimension as test static (***Figure 4.48***).

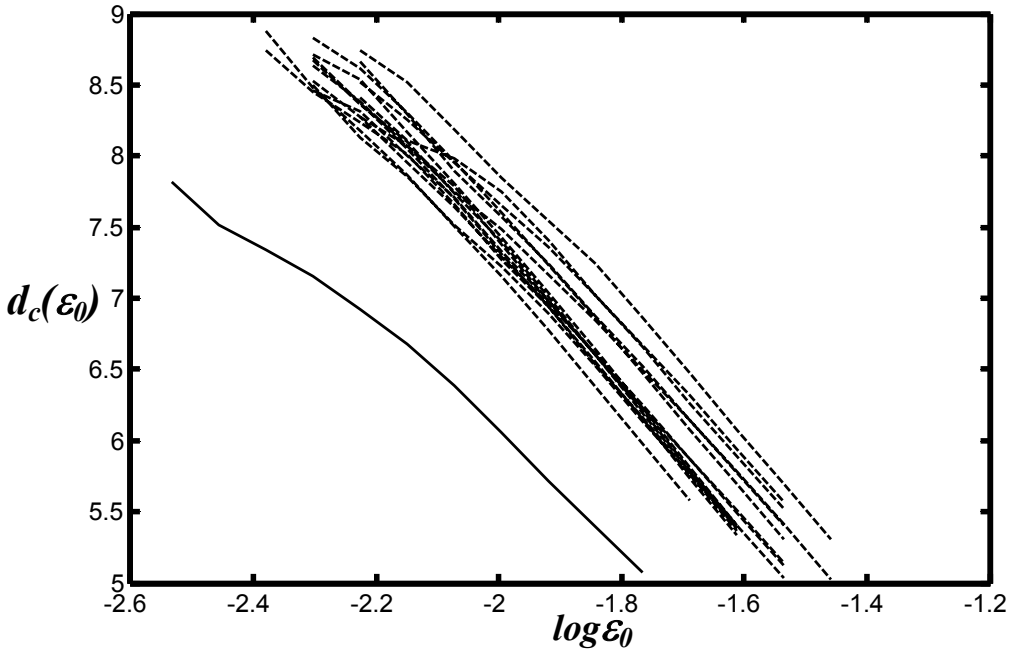


Figure 4.48: *Correlation dimension curves of surrogate and actual metal leaching data*

There is a definite separation between the curve of the actual data and that of the surrogate data sets. This suggests that the data could be deterministic, although the range of $d_c(\varepsilon_0)$-values of the curve from the actual data is larger than the embedding dimension of the data (> 5). If there is any determinism in the data, it is of a higher order. Noise in the data increases the dimensionality of an attractor, which can be the case here. The process data therefore lies in a grey area: not entirely deterministic, but also not entirely stochastic.

It is also important to note that Judd's algorithm for calculating the correlation dimension gets increasingly inaccurate for $d_c(\varepsilon_0)$-values above ±4 as result of a built in bias (Judd, 1992). When used as a pivotal statistic to compare data sets (as is done

with the surrogate data analysis) it can still be used with great success, since one is only interested in the relative differences.

4.3.3 Modelling the Metal Leaching Data

Before even attempting to model the data, one can already identify two issues that will complicate matters. One is the weak determinism in the data and the other is the lack of a long enough data set (only 1747 observations in the time series). The lack of data will have a severe impact on the ability of the model-based detection technique to give meaningful results

The time series is divided into four segments, each containing about 450 data points. A model is fitted to the first 450 data points of the time series. The model is then validated using data ("unseen") from the other three segments. It should be clear that this approach is fundamentally limited, as there are only three segments from which results can be obtained. One can also not afford splitting up the data set into more segments, as 450 data points are already pushing the limit of getting statistically credible results.

A MLP neural network model with 2 hidden nodes, as determined by the SIC (***Figure 4.49***), is used. According to the SIC an increase in the number of hidden nodes will not improve the model much. This is evident from the horizontal line in ***Figure 4.49***.

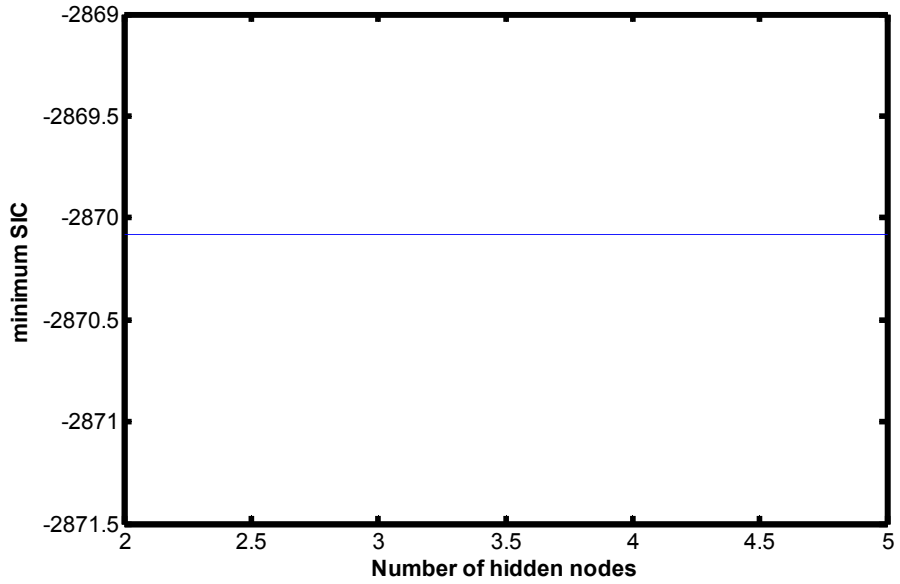


Figure 4.49: *SIC for modelling the metal leaching data.*

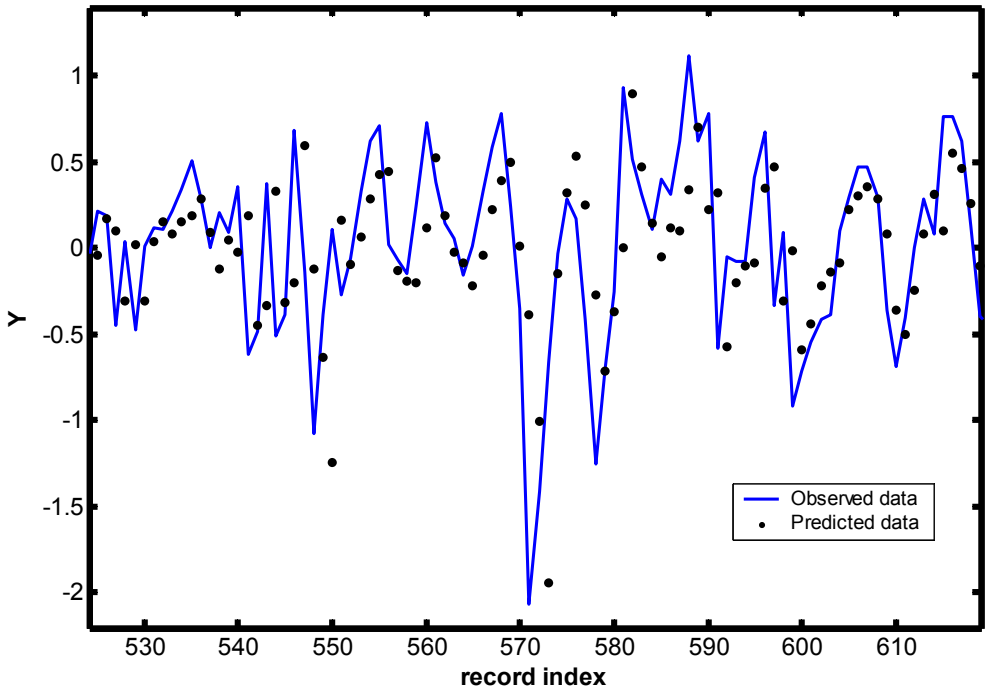


Figure 4.50: *Predicted and observed values for on-step prediction of metal leaching data.*

The one step prediction model fit is shown in ***Figure 4.50*** for a few of the data points. The R^2 values for the one-step prediction for each of the three segments are given in ***Table 4.2***. From these results, it seems as if the process was stationary in the first three segments, i.e. up to the first 1350 data points. After this point the R^2 statistic is much lower than for the previous segments. This *is* stretching it a bit, but from these results one must assume that the process parameters changed around the 1350th data point.

Table 4.2: *R^2 values for one-step prediction of metal leaching data.*

Segments		***R^2-statistic***
2	450 – 900	0.465
3	900 – 1350	0.426
4	1350 – 1747	0.250

4.3.4 Detecting Dynamic Change in the Metal Leaching Data Using Nonlinear Cross Prediction

The lack of a long enough time series prevents one from using this approach to its full potential. To do nonlinear cross predictions, segments containing at least a few hundred data points are needed (Schreiber, 1996). The length of this data set clearly rule out the use of mutual prediction maps, as used in the previous two case studies. The best one can do is to divide the time series into four segments containing 435 data points each, and then inspect the results from the cross prediction errors manually.

Table 4.3: *Cross prediction errors from the four segments of the metal leaching data.*

		DATA BASE SEGMENTS			
		1	2	3	4
PREDICTED SEGMENTS	1	0.8286	0.8376	0.8412	0.8583
	2	***0.8185***	***0.7946***	***0.8057***	***0.7868***
	3	0.875	0.8473	0.8205	0.8471
	4	***0.8834***	***0.8652***	***0.8764***	***0.8661***

The results from the nonlinear cross predictions are inconclusive. There are no consistency in the results from a change detection perspective. The only noticeable correlation in the results is that *segment 2* has the lowest prediction error when using *segments 1,2,3 and 4* respectively as data base, where *segment 4* has the highest error. This, however, does not reveal anything about the nonstationarity of the data set.

4.3.5 Using the Correlation Dimension to Detect Dynamic Change in Metal Leaching Data

This approach, which is based on comparing correlation dimension curves calculated from different parts of the time series, is even more challenging than the previous two methods (modelling and cross predictions) because of the limited amount of data. Already aware of the fact that Judd's implementation for calculating the correlation dimension needs at least 1000 points per segment to be reasonably reliable, having only 1700 observations available seems to eliminate this approach. The data can, however, be manipulated to overcome this 1000 point lower limit.

The first step in detecting nonstationarity using the d_c approach is to, as in the previous case studies, compare $d_c(\varepsilon_0)$-curves calculated from the two halves of the time series. To stick to the suggested lower limit of 1000 points, two overlapping segments are used: 1-1000 and 700-1700 (***Figure 4.51***). Although they contain some mutual information, most of the data from the segments are still data from the two halves, 1-850 and 850-1700.

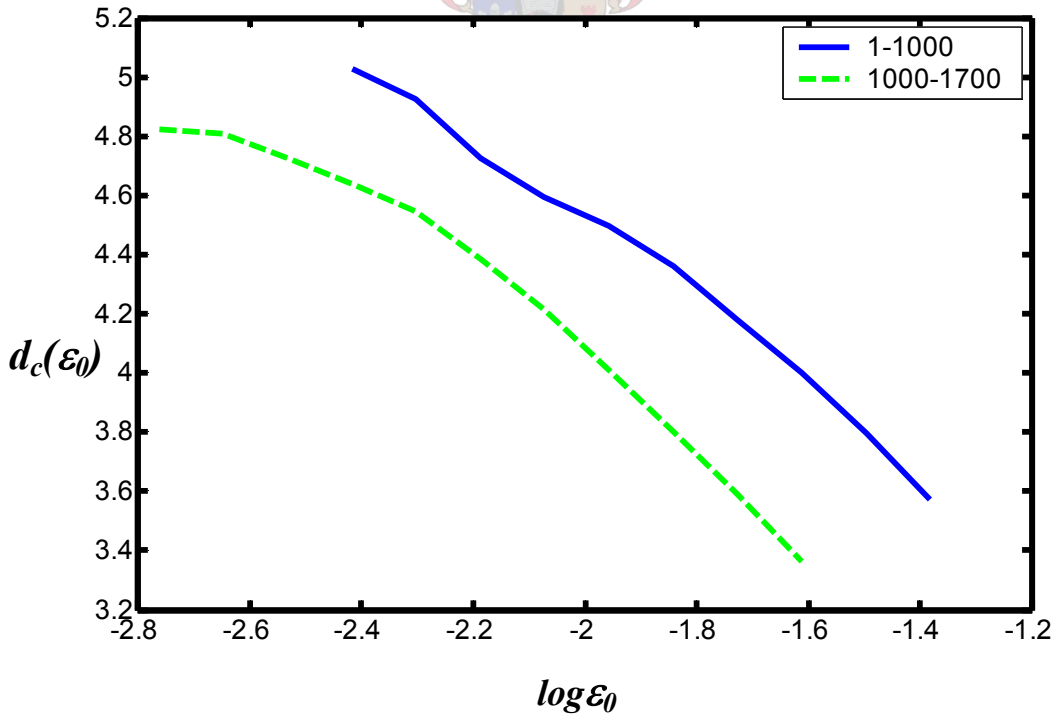


***Figure 4.51**: $d_c(\varepsilon_0)$-curves for the two halves of the metal leaching data.*

There is a definite dissimilarity between the curves from the two halves, which suggests that the data from the process is nonstationary. To obtain an estimate for the time at which the change took place, the moving window approach is applied. A window size of 800 data points is used, while moving the window forward every 100 data points (***Figure 4.52***). In doing this, using the suggested minimum of 1000 data points is rejected. The downside is that the results could be less reliable.

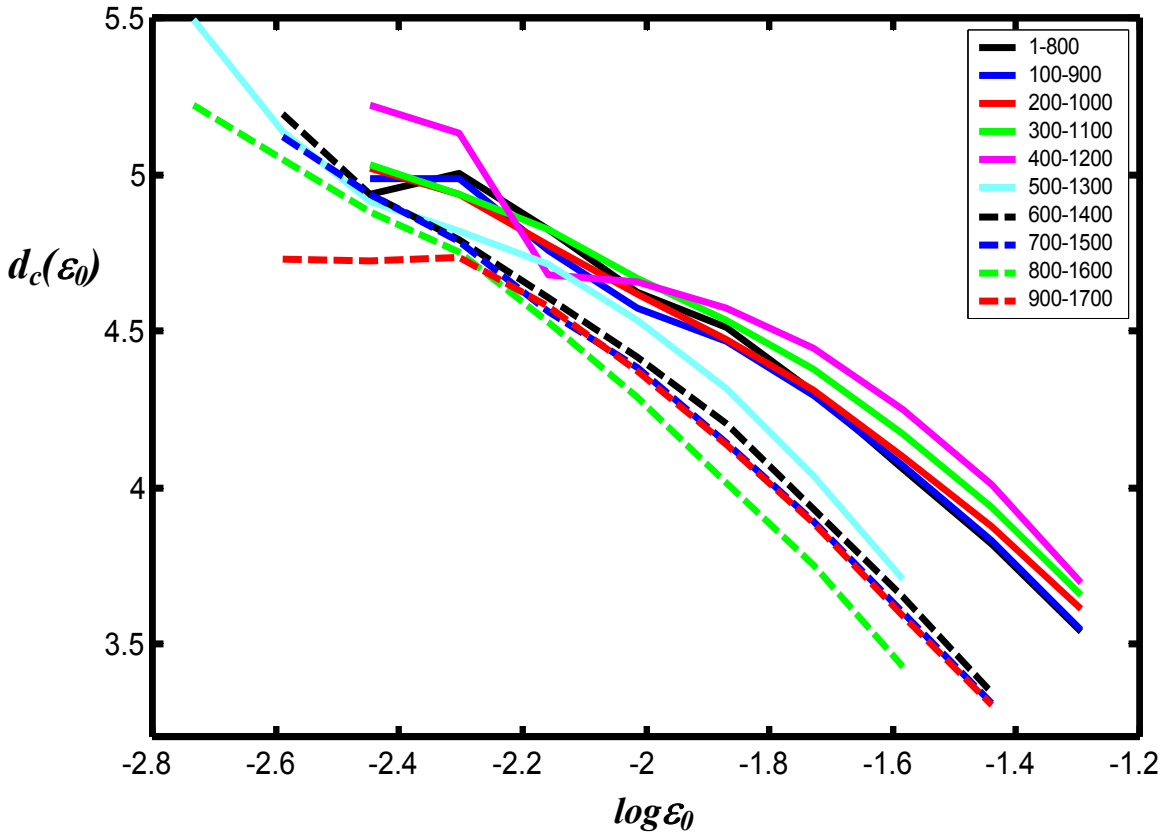


Figure 4.52: *$d_c(\varepsilon_0)$-curves from moving window of size 800 for metal leaching data.*

When examining ***Figure 4.52*** it can be seen that the $d_c(\varepsilon_0)$-curves from the *moving window segments 1-800* to *400-1200* are more or less similar. The dissimilarity of *segment 500-1300*, compared to the previous curves, suggests that the process parameters have changed. From *segment 600-1400* onwards the $d_c(\varepsilon_0)$-curves are similar once again. The conclusion one can make from the above results is that the

process parameters remained constant up to about data point 1200, after which the process parameters changed to a new value from data point 1300 onwards. This correlates with the results from the model-based detection, which indicated a parameter change at around data point 1350.

5 CONCLUSIONS

Techniques for reconstructing the state space of a system, from one-dimensional time series data, were successfully applied to three case studies. These include the average mutual information (AMI) statistic and autocorrelation function for determining the embedding delay; the false nearest neighbours (FNN) algorithm for estimating the embedding dimension; as well as singular spectrum analysis (SSA) whereby the embedding parameters are determined implicitly.

System classification by means of surrogate data analysis proved to be very useful for determining whether a process exhibits nonlinear deterministic or stochastic behaviour. In two of the case studies (the *autocatalytic reaction* and the *baker's map*) data were simulated from sets of nonlinear equations which are governed by strong deterministic rules. In both instances the systems were correctly classified as being nonlinear deterministic. Surrogate analysis of the data from the metal leaching plant suggested that only weak determinism was present. This was supported by the average results obtained when attempting to model the data.

Tests confirmed suspicions that model performance suffers when process data are nonstationary. If the process parameters stay relatively constant, the model predictions are quite good. The model performance, however, deteriorates when doing predictions in regions where there are dynamic changes. This model deterioration can serve as an indication for nonstationarity. An exception to the rule is when the attractor formed by the new set of process parameters, is embedded *within* the original attractor from which the model was estimated. The case study on the autocatalytic reaction highlighted this. Although the model fitness did decline somewhat, it could still give a good estimation to the new process dynamics.

A more formal way to detect and analyse nonstationarity in time series data is with nonlinear cross predictions. A plot of the mutual prediction errors is visually very informative and can give great insight into possible dynamical changes in the process.

This approach successfully detected “hidden” nonstationarities in the autocatalytic reaction and the baker’s map from time series data. No conclusive results, however, were obtained from the metal leaching plant data. The limited amount of data available for the analysis is most likely the main reason for the inconclusive results.

An assessment of the correlation dimension statistics, calculated from different segments of the time series, proved to be the most accurate method of detecting dynamic changes. Apart from positively identifying nonstationary in each of the case studies, it was also able to detect the parameter changes sooner than any other method tested. The correlation dimension approach is the best choice for online monitoring of changes in process dynamics, especially when used in a moving window framework.

The presence of measurement noise in the simulated data did not have a significant impact on the ability of the techniques to detect dynamic changes. This was verified by adding 10% measurement noise to data from the autocatalytic reaction.

The change detection approaches, as discussed in this work, are mainly limited to detecting changes in processes that have nonlinear dynamics with a dominant deterministic component. The amount of process data available for analysis is also a limiting factor, since the correlation dimension algorithm, and especially the nonlinear cross-prediction approach, are very data hungry. At least 2000 (or more) observations are needed for reliable detection of dynamic changes in a process.

6 REFERENCES

Addison, P.S. (1997). *Fractals and chaos.* London: IOP Publishing Ltd

Abarbanel, H.D.I (1993). The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics,* Vol. 65, No. 4, 1334-1392

Abarbanel, H.D.I (1996). *The analysis of observed chaotic data.* New York: Springer-Verlag

Abarbanel, H.D.I (1998). Obtaining order in a world of chaos. *IEEE Signal Processing Magazine,* May issue, 49-64

Azimi-Sadjadi, B & Krishnaprasad, P.S. (2002). Change detection for nonlinear systems; a particle filtering approach. *Proceedings of 2002 American Control Conference*, May 8-12, 4074-4079

Barnard, J.P. & Aldrich, C (2000). *Identifying nonlinear dynamic systems.* Datacube cc.

Barnard, J.P., Aldrich, C. & Gerber M. (2000). Identification of dynamic process systems with surrogate data methods.

Basseville, M. & Nikiforov, I. V. (1993). *Detection of abrupt changes: Theory and application*. Previously published by New Jersey: Prentice-Hall, Inc. [World Wide Web] Available: (http://www.irisa.fr/sigma2/kniga) [2002]

Bradly, E. (1996). *Time series analysis*. Unpublished thesis, University of Colorado, Colorado

De Oliveira, K. A., Vannucci, A. & Da Silva, E. C. (2000). Using artificial neural networks to forecast chaotic time series. *Physica A,* Vol. 284, 393-404

Eckman, J.P. & Ruelle, D. (1985), Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics,* Vol. 57, No. 3, 617-656

Fraser, A.M. & Swinney, H.L. (1986). Independent coordinates for strange attractors from mutual information. *Physics Review A,* Vol. 33, 1134-1140

Gershenfeld, N.A. & Weigend, A.S. (1993). *Forecasting the future and understanding the past*. MA: Addison-Wesley

Grassberger P. & Procaccia I. (1983). Characterization of strange attractors. *Physical Review Letters,* Vol. 50, 346-349

Haykin, S. (1999). *Neural Networks: A comprehensive foundation*. New Jersey: Prentice Hall, Inc.

Hegger, R., Kantz, H & Schreiber, T. (1998). *Practical implementation of nonlinear time series methods: The TISEAN package*. [World Wide Web] Available: *(http://www.mpipks-dresden.mpg.de/~ tisean).*

Hively, L. M., Gailey, P. C. & Protopopescu, V. A. (1999). Detecting dynamical change in nonlinear time series. *Physics Letters A,* Vol. 258, 103-114

Honkela, A. (2001). *Nonlinear switching state space models.* Unpublished masters thesis, Helsinki University of Technology, Helsinki

Judd, K. (1992). An improved estimator of dimension and some comments on providing confidence intervals. *Physica D,* Vol. 56, 216-228

Judd, K. & Mees, A. (1995). On selecting models for nonlinear time series. *Physica D,* Vol. 82, 426-444.

Juditsky, A., Hjalmarsson, H., Benviniste, A., Delyon, B., Ljung, L., Sjoberg, J. & Zhang, Q. (1995). Nonlinear Black-box models in System Identification: Mathematical Foundations. *Automatica,* Vol. 31, No. 12, 1725-1750

Kantz, H. & Schreiber, T. (1997). *Nonlinear time series analysis*. Cambridge: University Press

Kaplan, D. (1999). *Nonlinear function estimation, surrogate data, 1/f noise*. Notes for the 1999 IEEE-EMBS Summer School on advanced biomedical signal processing

Kennel, M.B. (1996). Statistical test for dynamical nonstationarity in observed time-series data. *Physical Review E*, Vol. 56, Nr.1, 316-321

Kennel, M.B., Brown, R. and Abarbanel, H.D.I. (1992). Determining minimum embedding dimension using a geometrical construction. *Physical Review A,* Vol. 45, 3403-3411

Kugiumtzis, D. & Christopherson, N. (1997). State space reconstruction: Method of delays vs singular spectrum analysis. *Research Report 236, University of Oslo, Oslo*

Lee, J.S. & Chang, K.S. (1996). Applications of chaos and fractals in process systems engineering. *Journal of Process Control,* Vol. 6, No. 2/3, 71-87

Levenberg, K. (1944). A method for the solution of certain nonlinear problems in least squares. *Quart. Applied Mathematics*, Vol. 2, 164-168.

Lorenz, E.N. (1963). Deterministic non-periodic flow. *J. Atmos. Sci.* Vol. 20, 130-141.

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

Marquardt, D. W. (1963). An algorithm for least squares estimation of nonlinear parameters. *Journal of SAIM*, Vol. 11, 431-441.

Nembhard, H.B. & Koa, M.S. (2002). *Adaptive forecast-based monitoring for dynamic systems*. To appear in *Technometrics*.

Owen, M (1989). *Statistical process control*. New York: Springer-Verlag

Palitz, U. (1995). Nonlinear time series analysis. *Proceedings of the NDES,* July 28-29, Dublin, England.

Perry, R.H. & Green, D.W. (1997). *Perry's Chemical Engineers' Handbook.* New York: McGraw-Hill

Sauer, T., Yorke, Y. & Casdagli, M. (1991). Embedology. *J. Stat. Phys.,* Vol. 65, 579-616

Schreiber, T. (1997). Detecting and analyzing nonstationarity in time series using nonlinear cross predictions. *Physical Review Letters*, Vol. 78, Nr. 5, 843-846

Schreiber, T. & Schmitz, A. (2000). Surrogate time series. *Physica D,* Vol. 142, 346-382.

Seborg, D.E., Edgar, T.F. and Mellichamp D.A. (1989). *Process dynamics and control*. New York: John Wiley & Sons

Statsoft, Inc.(1984), *Time series analysis*, [World Wide Web] Available: (http://www.statsoftinc.com/textbook/stathome.html) [2002]

Small, M. (1998). *Nonlinear dynamics in infant reparation.* Unpublished doctoral thesis, University of Western Australia,

Small, M. & Judd, K. (1998). Correlation dimension: A pivotal statistic for non-constrained realizations composite hypothesis is surrogate data analysis. *Physica D*, Vol. 120, 386-400.

Takens, F. (1981). *Detecting strange attractors in turbulence*. Lecture notes in Mathematics, Vol. 898, New York: Springer.

Theiler, J. (1995). On the evidence for low-dimensional chaos in an epileptic encephalogram. *Physics Letters A*, Vol . 196, 335-341.

Theiler, J. & Prichard, D. (1996). Constrained realization Monte Carlo method for hypothesis testing. *Physica D*, Vol. 94, 221-235.

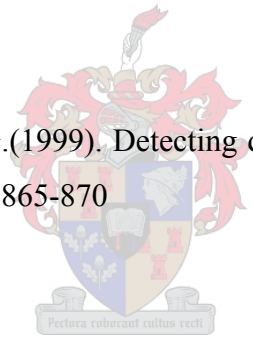
Theiler, J., Eubank, S., Longtin, A., Galdrikian, B & Farmer, J.D. (1992). Testing for non-linearity in time series: The method of surrogate data. *Physica D*, Vol. 58, 77-94.

Vining, G.G. (1998). *Statistical methods for engineers*. California: Brooks/Cole Publishing Company.

Willis, M.J. & Tham, M.T. (1994), *Advanced Process Control*. [World Wide Web] Available: (http://lorien.ncl.ac.uk/ming/advcontrl/sect1.htm) [2002]

Wise, B.M (1991).*Adapting multivariate analysis for monitoring and modelling of dynamic systems*. Unpublished doctoral thesis, University of Washington, Washington.

Yu, D., Lu, W. & Harrison, R.G.(1999). Detecting dynamical nonstationarity in time series data. *Chaos*, Vol. 9, Nr. 4, 865-870



APPENDIX

A. State Space Reconstruction Using Time Delay Embedding - A numerical Example

The time delay embedding method, whereby the internal dynamics of a system are reconstructed, can easily be explained with a numerical example. Suppose one has 300 observations from a process in the form of a time series (***Table A.1*** *and* ***Figure A.1***).

Table A.1: *Time series data*

s_n
8.5
9.0
10.0
11.3
13.0
14.9
17.1
19.1
20.7
21.5
…

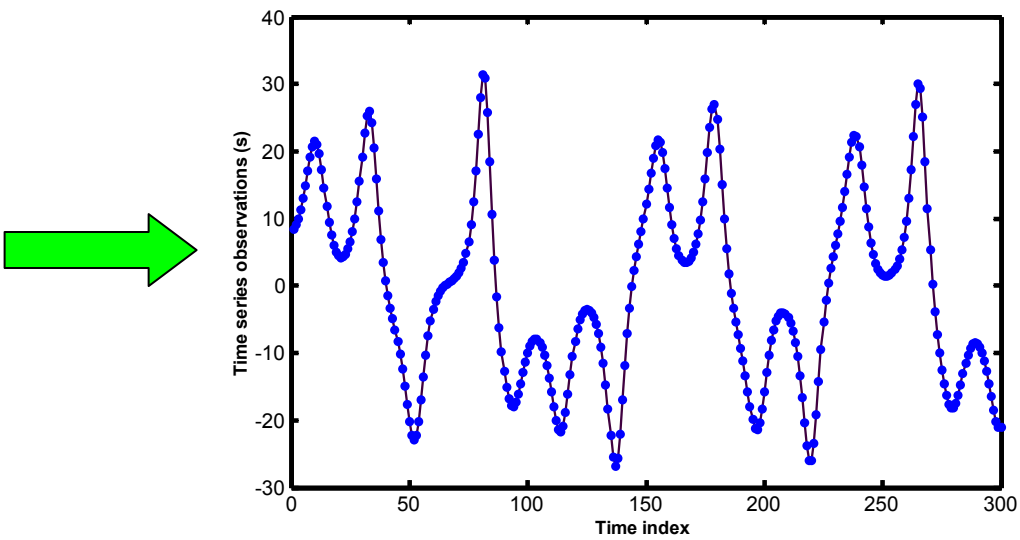


Figure A.1: *Plot of time series data*

Using the average mutual information (AMI) and false nearest neighbours (FNN) algorithms to determine the embedding parameters, one would find that a time delay (***l***) of 2 and an embedding dimension (***d***) of 3 will give an optimal embedding. Substituting these parameters into ***Equation 2.8,***

$$\boldsymbol{y_n} = (s_n, s_{n+l}, \ldots, s_{n+(d_e-1)l}), \tag{2.8}$$

yields the following reconstructed variables:

$$\boldsymbol{y_n} = (s_n, s_{n+2}, s_{n+4}). \tag{A.1}$$

The numerical equivalent is given in ***Table A.2***. An inspection of these numerical values, in addition to ***Equation A.1***, should clarify time delay embedding.

Table A.2: *Numerical values for the reconstructed variables*

s_n	s_{n+2}	s_{n+4}
8.5	10.0	13.0
9.0	11.3	14.9
10.0	13.0	17.1
11.3	14.9	19.1
13.0	17.1	20.7
14.9	19.1	21.5
17.1	20.7	21.1
19.1	21.5	19.6
20.7	21.1	17.2
21.5	19.6	14.5
…	…	…

The three-dimensional reconstructed attractor for the system can be seen in ***Figure A.2***.

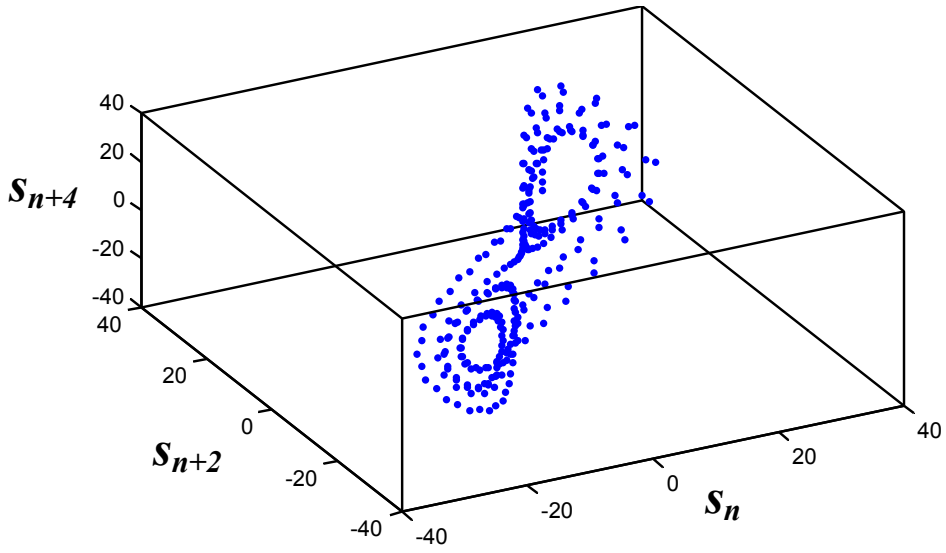


Figure A.2: *Reconstructed attractor for the numerical example*